

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Andrej Rolih

**Vgrajen sistem za avtomatizacijo
prostorov**

DIPLOMSKO DELO
VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: Viš. pred. dr. Robert Rozman

Ljubljana 2014

To delo je ponujeno pod licenco *Creative Commons Priznanje avtorstva-Deljenje pod enakimi pogoji 2.5 Slovenija* (ali novejšo različico). To pomeni, da se tako besedilo, slike, grafi in druge sestavine dela kot tudi rezultati diplomskega dela lahko prosto distribuira, reproducira, uporabljajo, priobčujejo javnosti in predelujejo, pod pogojem, da se jasno in vidno navede avtorja in naslov tega dela in da se v primeru spremembe, preoblikovanja ali uporabe tega dela v svojem delu, lahko distribuira predelava le pod licenco, ki je enaka tej. Podrobnosti licence so dostopne na spletni strani creativecommons.si ali na Inštitutu za intelektualno lastnino, Streliška 1, 1000 Ljubljana.



Izvorna koda diplomskega dela, njeni rezultati in v ta namen razvita programska oprema je ponujena pod licenco GNU General Public License, različica 3 (ali novejša). To pomeni, da se lahko prosto distribuira in/ali predeluje pod njenimi pogoji. Podrobnosti licence so dostopne na spletni strani <http://www.gnu.org/licenses/>.

Vsa izvorna koda, shema strojne opreme in ostale pripadajoče informacije so na voljo na:

<https://github.com/r00li/RhomeV3>

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Avtomatizacija bivalnih prostorov se s trenutno ponudbo komercialnih izdelkov in tovrstnih storitev še vedno nahaja v višjem cenovnem razredu. Tehnologija računalniških sistemov pa je medtem doživela intenzivni razvoj, ki se kaže v široki ponudbi cenovno dostopnejših in tehnološko naprednejših naprav oziroma sistemov. Izdelajte prototip takšnega cenovno dostopnega sistema s pomočjo naprav, ki jih najdete na trgu. Dopolnite sistem z lastnimi rešitvami tam, kjer je to potrebno in smiselno. Sistem naj zagotovi uporabniku udobno, prijazno in cenovno dostopno avtomatizacijo posameznega bivalnega prostora. Ob tem zagotovite tudi mrežno povezljivost in čim enostavnejše upravljanje sistema - tako lokalno kot na daljavo.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Andrej Rolih, z vpisno številko **63110109**, sem avtor diplomskega dela z naslovom:

Vgrajen sistem za avtomatizacijo prostorov

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom Viš. pred. dr. Robert Rozmana,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 16. septembra 2014

Podpis avtorja:

Kazalo

Povzetek

Abstract

1	Uvod	1
1.1	Predstavitev področja avtomatizacije	2
1.2	Avtorjev obstoječi domači sistem za avtomatizacijo	4
1.3	Zasnova vgrajenega sistema za avtomatizacijo	6
2	Vgrajen sistem za avtomatizacijo prostorov	9
2.1	Orodja in oprema	9
2.1.1	Mikrokrmnilnik STM32F4	10
2.1.2	Dodatna strojna oprema	12
2.1.3	Razvojno okolje Em::Blocks IDE	13
2.1.4	Orodje za načrtovanje vezij Cadsoft EAGLE	15
2.1.5	Realno časovni operacijski sistem FreeRTOS	16
2.2	Razvoj sistema	16
2.2.1	Vključitev FreeRTOS	17
2.2.2	Osnovne vhodno-izhodne naprave	18
2.2.3	Uporabniški vmesnik	19
2.2.4	Nadzor luči	23
2.2.5	Nadzor okenskih žaluzij	25
2.2.6	Vključitev dodatne strojne opreme	27
2.2.7	Načrtovanje in izdelava tiskanega vezja	28

KAZALO

2.2.8	Shranjevanje podatkov	33
2.2.9	Spletni strežnik in oddaljeno upravljanje	34
2.2.10	Daljinski upravljalnik in gumbi	36
2.3	Predstavitev izdelanega sistema	38
3	Sklepne ugotovitve	47

Seznam uporabljenih kratic

kratica	angleško	slovensko
OS	operating system	operacijski sistem
PWM	pulse-width modulation	pulzno-širinska modulacija
SRAM	static RAM	statični RAM
IDE	integrated development environment	razvojno okolje
PLC	programmable logic controller	programabilni logični krmilnik
HVAC	heating ventilation and air conditioning	ogrevanje, hlajenje in prezračevanje
HTTP	hypertext transfer protocol	protokol za prenos hiperteksta

Povzetek

Avtomatizacija prostorov je v današnjih časih hitro razvijajoča se panoga računalništva in elektronike. V ponudbi podjetji najdemo bogato izbiro izdelkov za avtomatizacijo. Sem sodijo celotni sistemi za nadzor in upravljanje zgradb kot tudi posamezne luči, ki jih lahko povežemo na internet. Ti izdelki uporabnikom omogočajo lažje bivanje in znižujejo stroške bivanja z nižanjem porabe energije. V tej nalogi je predstavljen postopek izdelave preprostega sistema za upravljanje posameznih prostorov, ki se osredotoča predvsem na lajšanje bivanja uporabnika. Izdelan sistem omogoča lokalno in oddaljeno (preko izdelanega spletnega vmesnika) upravljanje luči in okenskih žaluzij v prostoru. Hkrati naloga prikazuje, kaj vse lahko na področju avtomatizacije prostorov dosežemo z majhnim vložkom denarja.

Ključne besede: avtomatizacija, pametna hiša, vgrajen sistem, ARM.

Abstract

Home automation is a fast developing field of computer science and electronics. Companies are offering many different products for home automation. Ranging anywhere from complete systems for building management and control, to simple smart lights that can be connected to the internet. These products offer the user greater living comfort and lower their expenses by reducing the energy usage. This thesis shows the development of a simple home automation system that focuses mainly on the enhancement of living comfort. We have developed a system that can control lighting and window blinds through a local or remote web interface. It also shows what can be achieved with an investment of a small amount of money.

Keywords: automation, smart home, embeded system, ARM.

Poglavje 1

Uvod

Zamislimo si Janeza, ki pride po napornem zimskem dnevu domov iz službe. Vrata se mu samodejno odklenejo, ko se jim približa. Ko vstopi se avtomatično prižgejo luči na hodniku, hiša pa je že prijetno topla, saj se je ogrevanje vklopilo, ko je bil na poti domov. Ko se Janez usede v naslonjač se mu avtomatično prižge TV sprejemnik in preklopi na njegov priljubljeni kanal, luči v prostoru se avtomatično zatemnijo. Ko Janez zapusti prostor in odide v kuhinjo se TV in luči avtomatično ugasnejo. Janez začne pripravljati večerjo in, ko konča, hrano položi v pečico. Odide v delovno sobo, kjer preveri svoja elektronska sporočila in začne delo na enem izmed svojih projektov. Čez nekaj časa ga računalnik opozori, da naj v kuhinji preveri, kako je s hrano v pečici.

Take scenarije pogosto srečamo v znanstveno fantastičnih filmih in knjigah. Prikazujejo nam vizijo življenja v prihodnosti - nekaj, kar si želi velika skupina ljudi. Ti scenariji nam razkrivajo cilje hišne avtomatizacije. Malo ljudi se zaveda, da so podobni scenariji že danes uresničljivi in niso le del znanstvene fantastike.

Področje avtomatizacije prostorov in zgradb je zelo obširno. V današnjem času tehnologija avtomatizacije ni več omejena le na hiše bogatašev in velike poslovne objekte. Kaj avtomatizacija prostorov sploh je, in kake rešitve poznamo si lahko preberete v tem poglavju. Temu sledi kratek opis sistema

izdelanega v tej nalogi in povod za njegov nastanek.

1.1 Predstavitev področja avtomatizacije

Avtomatizacijo prostorov lahko v grobem razdelimo na dva dela: avtomatizacijo poslovnih objektov in avtomatizacijo doma. V primeru obeh poizkušamo avtomatizirati naslednja področja bivanja [1, 2, 3]:

- ogrevanje, hlajenje in prezračevanje (angl. *HVAC*)
- osvetljevanje
- senčenje
- varovanje
- multimedijske vsebine
- nadzor porabe energije

Pogosto poleg zgornjih področij bivanja dodajamo še domačo robotiko (angl. *domestic robotics*) in domače zdravljenje (angl. *telecare*), ki nekako dopolnjujeta zgornja področja. Ti dve področji sta še posebno pomembni za pomoč bolnim in starejšim na domu, saj lahko tako povečamo njihovo samostojnost. To je tudi eden izmed načinov, kako lahko omilimo posledice staranja prebivalstva.

V obeh primerih avtomatizacijo izvajamo zaradi dveh glavnih razlogov - prihranka na stroških in lajšanju bivanja. Stroške lahko zmanjšamo tako, da pametno izklapljamo luči in ostale naprave kadar v prostoru ni ljudi, izklopimo ali zmanjšamo ogrevanje kadar prostor ne bo v uporabi dalj časa, nadziramo porabo električne energije, pametno zasenčimo prostor v sončnih dneh in tako prihranimo pri stroških za hlajenje, in še veliko več.

Na obeh področjih lahko vidimo velik razcvet novih sistemov in standardov za avtomatizacijo, ki poizkušajo povezati naprave različnih proizvajalcev. Tu je verjetno dobro omeniti standard KNX, ki se je razvil iz več starejših

standardov [2] in poizkuša zagotoviti združljivost med napravami večih proizvajalcev. Standard KNX ponuja rešitve za avtomatizacijo poslovnih zgradb in avtomatizacijo doma. Omogoča nadzor vseh zgoraj omenjenih področij avtomatizacije. Težava protokola je le v potrebi po polaganju novih inštalacij (v primeru vgradnje v obstoječe prostore) in dragih napravah, ki jih v sistem lahko povežemo. Prvo težavo rešuje brezžična različica protokola medtem, ko se bo cena naprav spustila le ob bolj množični uporabi protokola.

Avtomatizacija poslovnih stavb se uporablja že dolgo let. Sem sodijo zahtevni centralni sistemi, ki nadzorujejo različna zgoraj naštetá področja skozi celotno stavbo. Avtomatizacija stavb se je začela s pnevmatskim nadzorom ogrevanja [6]. Taki sistemi so bili priljubljeni že v šestdesetih letih prejšnjega stoletja in se pogosto uporabljajo še danes. Moderni sistemi k temu dodajajo vedno več področji, ki jih nadzorujejo. Za upravljanje se pogosto uporabljajo programabilni logični krmilniki (angl. *PLC - Programmable Logic Controller*) v kombinaciji z običajnimi računalniki. Taki krmilniki so običajno namenjeni posebej avtomatizaciji prostorov in se odlikujejo z visoko zanesljivostjo. Posledica tega je tudi visoka cena tako izdelanih sistemov.

Na drugi strani je področje avtomatizacije doma, katerega del je tudi sistem opisan v tej nalogi. Tu je bil razvoj avtomatizacije veliko počasnejši. Centralni sistemi za avtomatizacijo (omenjeni zgoraj) so še danes redki in v večini primerov, omejeni na hiše bolj premožnih ljudi. Kljub temu se v zadnjih letih vgrajuje čedalje več takih sistemov. V ZDA je bilo leta 2012 nameščenih 1.5 milijonov sistemov za avtomatizacijo domov [7] kar je skoraj dvakrat več kot v letu 2011. Pričakuje se, da bo leta 2017 nameščenih kar 8 milijonov novih sistemov za avtomatizacijo. Še vedno pa taki sistemi niso lahko cenovno dostopni. Na tem področju so zato zelo popularne domače individualne rešitve različnih zanesenjakov, ki so doživele velik razcvet v zadnjih letih.

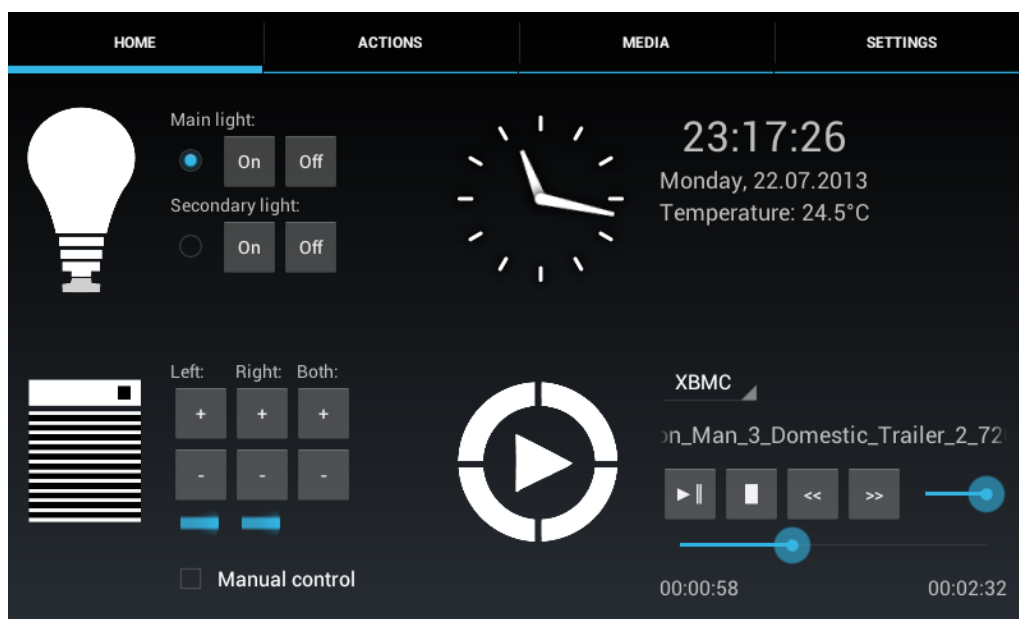
Z razvojem mobilnih naprav (pametnih telefonov in tablic) lahko vidimo tudi premik iz centraliziranih sistemov v posamezne avtomatizirane naprave. Tako so v zadnjih letih popularne različne naprave za avtomatizacijo, ki jih

lahko nadzorujemo z mobilnimi napravami. Sem sodijo pametni termostati (na primer Nest [8]), pametne luči in žarnice (na primer Philips Hue [9]), pametne klimatske naprave, in še veliko drugih. Taki sistemi imajo običajno nizke cene in so enostavni za vgradnjo. Te sisteme lahko namesti uporabnik sam, brez potrebe po zahtevnem polaganju inštalacij. Še ena prednost sistemov iz te skupine je, da jih lahko uporabnik vgrajuje in nadgrajuje postopoma. Imajo pa ti sistemi tudi slabosti. Pri njih nimamo možnosti centraliziranega upravljanja, pogosto se zanašajo na lastniške zaprte protokole in hitro postanejo zastareli, ker ne omogočajo nadgrajevanja.

Poleg centralnih sistemov in posameznih avtomatiziranih naprav pa poznamo še nekakšno kombinacijo obeh. Ti sistemi omogočajo centralno upravljanje naprav preko mobilnih aplikacij, vgradnih tipk ali ostalih naprav. Sestavljajo jih različni moduli za prižiganje luči, senzorji gibanja, nadzorne kamere, ... V to skupino spadajo protokoli, kot so X10 [10], Insteon [11], Controlled Comfort [12] in drugi. Ti protokoli so v večini primerov lastniški in omejeni na enega ali nekaj proizvajalcev. Prav tako ne ponujajo vseh funkcij centraliziranih sistemov. Namesto tega ponujajo nizko ceno in preprosto modularno vgradnjo. Pogosto za komunikacijo med napravami uporabljajo katero od tehnologij za komunikacijo preko visokonapetostnega omrežja ali katero od tehnologij za brezžično povezovanje. To omogoča enostavno vgradnjo v že obstoječe prostore in omogoča postopno avtomatizacijo posameznih prostorov.

1.2 Avtorjev obstoječi domači sistem za avtomatizacijo

Kljub različnim rešitvam za hišno avtomatizacijo, ki jih najdemo na trgu so te še vedno neprilagodljive in drage. Pogosto tudi ne rešujejo težav, ki bi si jih uporabniki želeli rešiti. V zadnjih letih se je zato zelo razcvetel trg domačih sistemov za avtomatizacijo. Razlog za ta razcvet je verjetno vse večja dostopnost različnih razvojnih platform. Te so po navadi poceni in



Slika 1.1: Uporabniški vmesnik sistema RHome

enostavne za uporabo. Tukaj je zagotovo vredno omeniti platformi Arduino in Raspberry Pi, ki sta botrovala k nastanku veliko različnih sistemov za avtomatizacijo.

Avtor te naloge je v preteklosti že izdelal lasten sistem za avtomatizacijo prostorov. Slika 1.1 prikazuje uporabniški vmesnik tega sistema. Sistem sestavljata mikrokrmilniški sistem Arduino z nekaj lastne strojne opreme in tablica Android nameščena na steno, ki skrbi za nadzor in upravljanje sistema. Sistem omogoča naslednje:

- nadzor luči
- nadzor okenskih žaluzij
- nadzor naprav z IR sprejemniki (TV, zvočniški sistem, igralna konzola, ...)
- nadzor medijskega predvajalnika XBMC [13]
- nadzor sistema z daljinskim upravljalnikom

Ta sistem že več kot eno leto uspešno skrbi za nadzor avtorjeve sobe. Vendar ga pesti nekaj težav, ki so vodile v nastanek izboljšanega, vgrajenega sistema opisanega v tej nalogi.

Prejšnji sistem se namreč zanaša na tablico, ki jo poganja Android. To pomeni, da brez tablice sistem ne deluje. Potreba po delujoči Android tablici tako močno dvigne ceno takega sistema. Druga težava je tudi nepraktičnost sistema. Tablica mora biti tako neprestano pod zunanjim napajanjem in hkrati preko kabla povezana na Arduino. Ta mora zaradi kontrole okenskih žaluzij uporabljati ločen napetostni vir, saj USB povezava ni zmožna dovajati dovolj energije za pogon motorjev. V trenutni izvedbi prav tako ne omogoča oddaljenega upravljanja kar pa bi bilo z nekaj popravki programske opreme enostavno dodati.

1.3 Zasnova vgrajenega sistema za avtomatizacijo

Vse zgoraj opisane težave so ustvarile potrebo po nečem boljšem. Nekaj kar bi bilo enostavno in poceni podvojiti in namestiti tudi v druge prostore. Zaželeno je tudi možnost enostavne nadgradnje in nadaljnjega razvoja sistema.

Kot pri prejšnjem sistemu je osnovna ideja še vedno v razpršenem upravljanju. Osnovna enota avtomatizacije je tako posamezen prostor. To odpravi potrebo po namestitvi dragih inštalacij in omogoča postopno uvajanje sistema. Prav tako poskrbi za boljšo zanesljivost sistema. V primeru napake na sistemu lahko enostavno zamenjamo, ali popravimo okvarjen del, medtem, ko bodo sistemi v ostalih prostorih še naprej nemoteno delovali.

Kot pri prejšnjem sistemu je tudi tukaj najpomembnejša zahteva udobje. Po mnenju avtorja te naloge se današnji sistemi za avtomatizacijo vse preveč osredotočajo na nadzor in zmanjševanje porabe energije. To sisteme sicer naredi uporabne, vendar ne poveča uporabnikovega udobja. To je po mnenju avtorja tudi eden izmed razlogov za slabo razširjenost sistemov za

avtomatizacijo. Nadzor porabe energije je tako za nov sistem le možnost za razširitev in ne osnovna naloga.

Velika večina pomanjkljivosti prejšnjega sistema se skriva v potrebi po tablici za nadzor. V kolikor tablico odstranimo in v sistem vključimo lokalni vmesnik za upravljanje bomo, tako odpravili skoraj vse težave prejšnjega sistema. Še najbolj pa se bo znižala cena.

Še ena izmed težav prejšnjega sistema je omejenost na le en prostor. Prejšnji sistem je namreč narejen le za specifično sobo. Vsaka sprememba (kot je na primer dodajanje luči) potrebuje spreminjanje kode. Prav tako je vezan na strojno opremo tablice, saj uporabniški vmesnik podpira le eno ločljivost zaslona. V obeh primerih je problem mogoče rešiti programsko, vendar bi to zahtevalo kar nekaj dela, zato je bolje začeti znova z boljše zasnovanim sistemom.

Nov sistem naj bi tako omogočal naslednje:

- nadzor luči
- nadzor okenskih žaluzij
- oddaljeno upravljanje sistema (spletni strežnik)
- upravljanje sistema preko lokalnega uporabniškega vmesnika v sobi (nadmestitev tablice z integriranim zaslonom in vhodnimi napravami)
- upravljanje sistema preko daljinskega upravljalnika
- večja modularnost in prilagodljivost (ni potrebe po spreminjanju kode ob spremembah v sobi)
- upravljanje medijskega predvajalnika XBMC
- zaznavanje prisotnosti

Poglavje 2

Vgrajen sistem za avtomatizacijo prostorov

Izdelava sistema se je začela z določitvijo osnovnih zmožnosti, ki jih bo sistem ponujal. Opis teh se nahaja v uvodu te naloge. Izbiri želenih zmožnosti sistema je sledila izbira ustreznih orodij in opreme za izdelavo sistema.

Nov sistem deluje kot samostojna enota in za delovanje ne potrebuje dodatne opreme. Odlikuje ga visoka zmogljivost, prilagodljivost in enostavnost uporabe. Hkrati je tudi izjemno kompakten (zasede le nekoliko večjo površino od povprečnega zidnega stikala) kar ga naredi primerne za vgradnjo skoraj kamorkoli v prostor.

V naslednjem podpoglavju sledi opis uporabljenih orodij in opreme. Temu sledi opis in natančna predstavitev izboljšane sistema, ki je bil izdelan v tem delu.

2.1 Orodja in oprema

Za izdelavo vgrajenega sistema poleg običajnih programskih orodij potrebujemo tudi nekaj strojne opreme in storitev. Pri razvoju vgrajenih sistemov je izbira ustrezne strojne opreme zelo pomembna. Ta namreč pogojuje tudi izbiro programskih orodij, ki jih bomo uporabili.

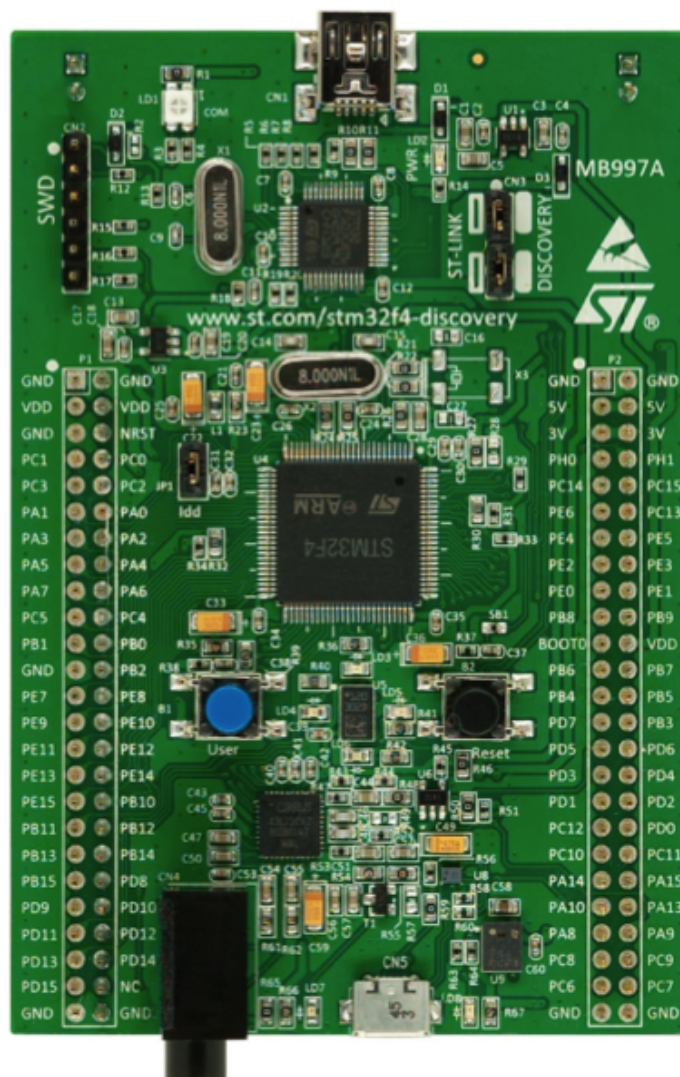
2.1.1 Mikrokrmilnik STM32F4

Mikrokrmilnik (lahko pa tudi mikroprocesor ali sistem na čipu) je danes najbolj pomemben sestavni del vsakega vgrajenega sistema. V preteklosti je bilo za izdelavo vgrajenega sistema potrebno veliko podpornega vezja (angl. *glue logic*). Danes moderni mikrokrmilniki vsebujejo skoraj vse potrebno za delovanje. Sem spadajo različni krmilniki za komunikacijo, analogno-digitalni pretvorniki, časovniki, enote za računanje s plavajočo vejico, ... Vse te vgrajene naprave omogočajo hitrejši, lažji in cenejši razvoj sistemov. Tako lahko lasten vgrajen sistem razvije skoraj vsakdo brez potrebe po dragi opremi in dolgoletnih izkušnjah.

Za razvoj se je kot možen kandidat za izdelavo najprej ponudila odprta platforma Arduino [4]. Arduino je strojna in programska platforma za hiter razvoj vgrajenih sistemov. Odlikuje jo nizka cena in enostavnost uporabe. Za strojno osnovo uporablja predvsem 8-bitne mikrokrmilnike AVR podjetja Atmel. A za sistem opisan v tej nalogi se je hitro izkazalo, da strojne zmogljivosti platforme ne ustrezajo zahtevam.

Tako je bila namesto Arduina izbrana platforma ARM. ARM je platforma istoimenskega podjetja namenjena razvoju vgrajenih sistemov za skoraj kakršnokoli področje. Izdelke podjetja ARM najdemo v mobilnih telefonih, tablicah, televizijskih sprejemnikih, gospodinjskih aparatih, ... ARM kot podjetje ne izdeluje strojne opreme ampak le načrte procesorskih jeder in orodja za izdelavo procesorjev in mikrokrmilnikov. Te načrte lahko druga podjetja odkupijo in z njihovo pomočjo naredijo svoj sistem. Druga podjetja temu jedru v večini primerov dodajo še druge enote in mikrokrmilnik ali procesor tudi fizično izdelajo.

Za nalogo je bil tako izbran mikrokrmilnik STM32F4 podjetja STM, zgrajen na arhitekturi ARM Cortex M4. Cortex M4 je najzmogljivejša ARM arhitektura za mikrokrmilnike [5] namenjena predvsem hitremu procesiranju digitalnih signalov. Za razvoj je bila uporabljena razvojna ploščica STM32F4 Discovery (prikazana na sliki 2.1) katere glavni del je mikrokrmilnik STM32F407VG. Nekaj pomembnih tehničnih podrobnosti:



Slika 2.1: Razvojna ploščica STM32F4 Discovery

- procesor s frekvenco 168MHz
- 1MB vgrajenega pomnilnika flash
- 128kB statičnega RAM pomnilnika (+64kB, ki se s pomočjo zunanje baterije ohranja ob izgubi napajanja)
- 4 vmesniki USART
- 3 vmesniki SPI
- ura realnega časa (angl. *Real Time Clock*)
- 17 časovnikov

Razvojna ploščica vključuje tudi vmesnik USB in vgrajen vmesnik za programiranje kar poskrbi, da za začetno delo druge strojne opreme ne potrebujemo.

Na odločitev je pomembno vplivalo dejstvo, da ima mikrokrmilnik vgrajen velik flash pomnilnik. To je pomembno saj za izdelavo kompleksnega uporabniškega vmesnika in vključitev spletnega strežnika potrebujemo dovolj prostora za programsko kodo. Hiter procesor pa omogoča, da lahko mikrokrmilnik hitro reagira na različne podatke iz senzorjev in hkrati izrisuje uporabniški vmesnik na zaslon in gosti spletno stran za upravljanje. Kljub svoji zmogljivosti pa ga odlikuje relativno nizka cena in enostavna uporaba. Za delovanje namreč potrebuje le malo dodatnih elektronskih komponent. S to razvojno ploščico smo delali tudi na fakulteti v sklopu predmeta vgrajeni sistemi, kar je pomagalo skrajšati čas razvoja.

2.1.2 Dodatna strojna oprema

Poleg mikrokrmilnika je bilo za izvedbo projekta potrebno izbrati še nekaj druge strojne opreme.

Zelo pomemben je zaslon LCD saj predstavlja glavni vmesnik za komunikacijo z uporabnikom. Zaradi cene je bil izbran preprost barvni zaslon velikosti 2,2 palca z ločljivostjo 320x240 pik, ki ima vgrajen krmilnik ILI9341

podjetja Ilttek. Krmiljenje poteka preko serijskega vmesnika SPI, kar zmanjša kompleksnost sistema, saj je za komunikacijo potrebnih manj povezav.

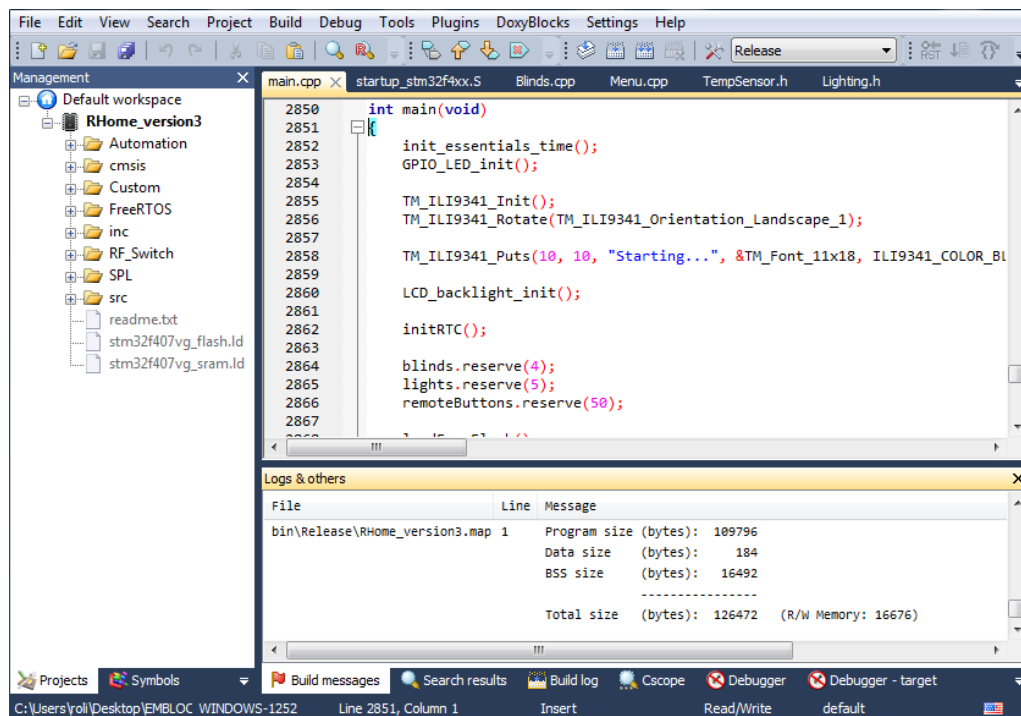
Naslednja pomembna komponenta je modul WiFi za dostop do omrežja in interneta. Izbran mikrokrmilnik že ima vgrajen vmesnik Ethernet, vendar pa je bilo za sistem bolje, če se lahko poveže brez uporabe žic. Tako nam pri namestitvi sistema ni potrebno polagati kablov za omrežje. Zaradi svoje zelo nizke cene je bil uporabljen modul HLK-RM04 proizvajalca Hi-Link. Ta s krmilnikom komunicira preko vmesnika USART. Sam modul pa lahko deluje kot TCP/UDP strežnik ali odjemalec. To pomeni, da programerju sistema ni potrebno skrbeti za nižje plasti TCP/IP modela ampak lahko neposredno uporablja višje plasti, kar močno poenostavi razvoj.

Za brezžično komunikacijo z lučmi, ostalo hišno napeljavo in daljinskim upravljalnikom je bil uporabljen generični 433MHz radijski vmesnik. S pomočjo tega lahko preko različnih protokolov komuniciramo z veliko izbiro brezžičnih stikal, ki jih najdemo v trgovinah.

Sistemu so dodani še rotacijski kodirnik (angl. *rotary encoder*), LED diodi in nekaj gumbov za potrebe upravljanja s sistemom. Dodana pa sta še sprejemnik in oddajnik IR ter vmesnik Bluetooth LE (low energy). Ti sicer zaradi časovnih omejitev niso bili uporabljeni v nalogi, vendar so prisotni na izdelanem tiskanem vezju in ponujajo možnost nadaljnjega razvoja.

2.1.3 Razvojno okolje Em::Blocks IDE

Za delo na projektu je bilo uporabljeno razvojno okolje Em::Blocks. Gre za nekoliko manj znano odprtokodno okolje za razvoj programske opreme za vgrajene sisteme. Temelji na odprtokodnem Code::Blocks, ki je namenjen razvoju programske opreme za osebne računalnike v jezikih C in C++. Izbira pravega okolja za razvoj na ARM Cortex arhitekturi je lahko zelo težavna. Za razliko od programskih orodij namenjenih razvoju programske opreme za računalnike so taka okolja veliko bolj redka in pogosto plačljiva. Cene licenc zanje tako pogosto presegajo znesek 1000EUR. Prav tako so v večini primerov namenjena večjim podjetjem in ne domačim uporabnikom.



Slika 2.2: Razvojno okolje Em::Blocks IDE

Študentje in domači uporabniki so tako omejeni na uporabo omejenih različic teh okolij, ki imajo v veliki večini primerov omejitev velikosti kode ter ne ponujajo nekaterih funkcij. Lahko se odločijo tudi za katerega izmed orodij za delo preko ukazne vrstice, ki so po navadi zahtevna za uporabo, ali izberejo katerega izmed orodij, ki temeljijo na razvojnem okolju Eclipse. Ta so v večini primerov cenejša ali brezplačna vendar počasna in včasih zahtevna za uporabo.

Em::Blocks odlikuje enostaven in preprost uporabniški vmesnik (kot prikazuje slika 2.2). Ponuja nam čarovnike za ustvarjanje novih projektov za veliko različnih procesorjev kar močno olajša delo začetnikom. Pri delu z vgrajenimi sistemi je namreč pred začetkom dela potrebno označiti v kateri del pomnilnika naj se shrani koda, koliko prostora ima pomnilnik, s kakšno frekvenco deluje procesor in še veliko več. Em::Blocks in druga dobra razvojna okolja nam tako prihranijo veliko časa. Prav tako Em::Blocks porabi

malo sredstev in je nadvse hiter kar ga naredi veliko bolj prijetnega za uporabo kot konkurenčni Eclipse.

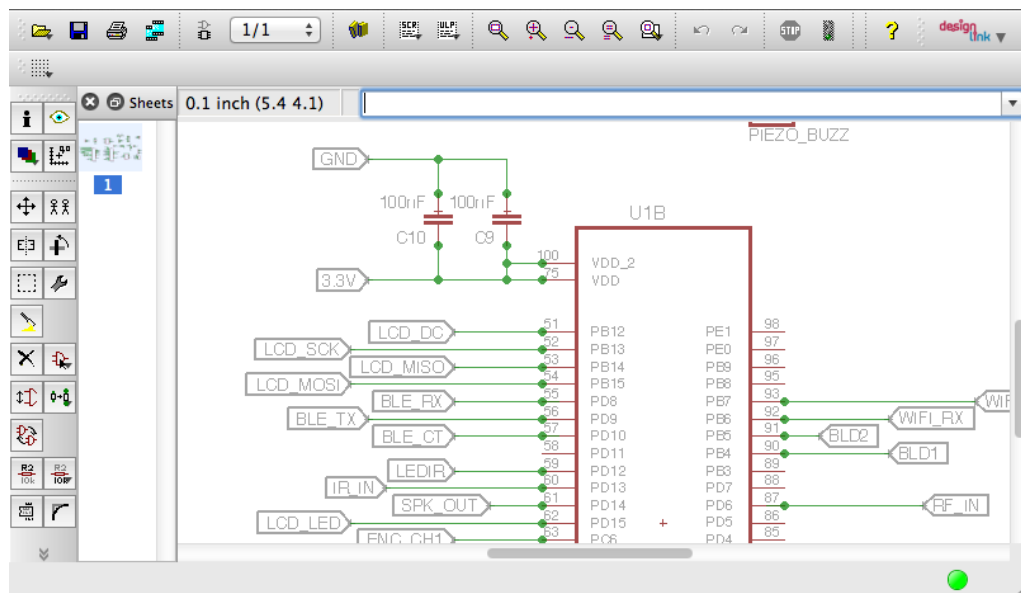
Za razvoj vgrajenih sistemov se velikokrat uporablja programski jezik C, saj omogoča pisanje kode, ki je hitra in brez nepotrebnega balasta. V izdelavi sistema je bil uporabljen programski jezik C v kombinaciji s C++. Ta je bolj primeren za delo z grafiko in vključuje kar nekaj dodatkov, ki omogočajo hitrejši in lažji razvoj. To za sabo prinese nekaj balasta, kar opazimo v nekoliko večji prevedeni kodi. Ker je bil za delo izbran zmogljiv mikrokrmilnik z veliko prostora za programsko kodo, izbira C++ ni predstavljala večje težave.

2.1.4 Orodje za načrtovanje vezij Cadsoft EAGLE

EAGLE je EDA (angl. *Electronic Design Automation*) orodje namenjeno načrtovanju električnih shem in risanju tiskanih vezij. Po končani razvojni stopnji je bilo namreč potrebno sistem za avtomatizacijo prenesti v bolj primerno obliko za uporabo. To pomeni, da je bilo potrebno izrisati in izdelati tiskano vezje, ki bo podpiralo vse komponente in bilo hkrati čim manjše.

Za izdelavo je bil uporabljen program EAGLE podjetja Cadsoft. EAGLE je med domačimi uporabniki zelo priljubljen, saj je enostaven za uporabo in ima hkrati veliko podpore in knjižnic s komponentami. Podjetje za domače uporabnike ponuja brezplačno licenco z nekaj omejitvami (omejena velikost izdelanega tiskanega vezja in največ dva signalna nivoja) kar ga naredi tudi cenovno ugodnega. Veliko podjetij, ki se ukvarjajo z izdelavo tiskanih vezij ima že pripravljene predloge za Eagle, kar olajša izris in izdelavo, saj nam ni potrebno ročno vnašati parametrov za generiranje proizvodnih datotek in ročno preverjati ali je narisano tiskano vezje skladno z proizvajalčevim proizvodnim procesom.

Slika 2.3 prikazuje EAGLE z odprtim delom sheme tukaj predstavljenega sistema.



Slika 2.3: Program EAGLE

2.1.5 Realno časovni operacijski sistem FreeRTOS

FreeRTOS je realno časovni operacijski sistem (angl. *Real Time Operating System - RTOS*) namenjen vgrajenim sistemom. Njegova implementacija je preprosta in porabi malo prostora, zato je namenjen sistemom z zelo omejenimi zmogljivostmi kot so mikrokrmilniki. Bolj kot pravi operacijski sistem ga lahko obravnavamo kot knjižnico za delo s procesi. Njegova glavna naloga je ponuditi možnost izvajanja večih procesov in preklapljanje med njimi. Na ta način lahko na primer omogočimo, da se sistem med risanjem na zaslon, (na videz) hkrati odziva tudi na pritiske tipk uporabnika. Dodaja tudi nekaj funkcij za lažje upravljanje s pomnilnikom in nekaj funkcij in podatkovnih struktur za usklajevanje procesov.

2.2 Razvoj sistema

Ko so bile naloge novega sistema določene in strojna ter programska oprema izbrana, se je lahko pričel razvoj. Vse se je začelo z ustvarjanjem novega pro-

jekta v razvojnem okolju in nastavitvijo nekaterih parametrov. Sem spadajo izbira želene frekvence procesorja, izbira programskega jezika in vključitev ustreznih sistemskih knjižnic. Za komunikacijo z napravami vključenimi v procesor (časovniki, SPI krmilniki in ostalo) je poskrbel STM kot proizvajalec mikrokrmilnika. Ta je, izdal ustrezne knjižnice, napisane v programskem jeziku C. Poleg knjižnic proizvajalca procesorja pod sistemske knjižnice spadajo tudi knjižnice, ki jih je pripravilo podjetje ARM. Sem spadajo knjižnice za delo s sistemskim časovnikom in prekinitvenim krmilnikom.

2.2.1 Vključitev FreeRTOS

Sledila je vključitev FreeRTOS in njegova ustrezna konfiguracija. To storimo z ustrezno izbiro parametrov v eni izmed konfiguracijskih datotek. Tukaj nastavimo, ali želimo uporabljati prekinitve za preklapljanje med opravili (angl. *preemption*) in na koliko časa naj se preklop med opravili izvede. Nastavimo tudi, ali želimo uporabljati nekatere posebne funkcije operacijskega sistema. Nastavimo še zeleno minimalno velikost sklada. Sledi ustvarjanje nekaj testnih procesov, ki utripajo na razvojno ploščico integrirane LED diode, s katerimi preverimo, ali OS ustrezno deluje.

Vsi procesi v sistemu so izvedeni na preprost način. Vsak proces je namreč ena funkcija. Ta funkcija lahko vsebuje svoje lokalne spremenljivke in mora vsebovati eno neskončno zanko, ki predstavlja proces. Vsakič ko zanka naredi en obhod (lahko) pokličemo funkcijo operacijskega sistema, ki trenutni proces zaustavi za zelen čas. Operacijski sistem bo tako začel z izvajanjem drugega procesa in se po pretečenem času vrnil nazaj. Procese seveda lahko prekinemo tudi sredi izvajanja, preden končajo en obhod zanke. Tako dosežemo navidezno hkratno delovanje več procesov.

Kasneje med razvojem dodamo vse potrebne procese za pravilno delovanje sistema. V skupnem dodamo 4 različne procese, ki izvajajo naslednje naloge:

- osveževanje glavnega zaslona
- spletni strežnik

- interakcija z uporabnikom (izris menijev in odzivanje na vhod)
- izvajanje zakasnenih akcij (način za spanje, avtomatski nadzor žaluzij) in odzivanje na daljinski upravljalnik

2.2.2 Osnovne vhodno-izhodne naprave

Ko je OS ustrezno preizkušen sledi povezovanje s prvim kosom strojne opreme. Za začetek je bil to LCD zaslon. Odločimo se, katere nožice na razvojni ploščici bomo izbrali za povezovanje zaslona. Na voljo imamo tri SPI vmesnike in vsakega izmed njih lahko programsko povežemo na več (vendar ne vse) nožic razvojne ploščice. Pri ustrezni izbiri nožic nam pomaga dokument proizvajalca, kjer so navedene funkcije vsake izmed nožic. Glede na shemo proizvajalca povežemo ustrezne priključke zaslona na izbrane nožice razvoje ploščice. Dodatno nožico porabimo za priklop osvetlitve zaslona. Tako lahko osvetlitev zaslona po želji izklopimo ali vklopimo programsko. Za krmiljenje zaslona je bila izbrana ena izmed odprtokodnih knjižnic za ta krmilnik [16] kar je omogočilo hitrejši razvoj. Knjižnica podpira osnovne grafične operacije. Na zaslon lahko izriše posamezno točko, besedilo, kvadrat, krog ali črto. Sledilo je krajše testiranje zmogljivosti zaslona. Ob tem opazimo, da je risanje na zaslon počasna operacija. Izris grafičnih elementov, ki niso besedilo tako traja kar nekaj časa. Krajše raziskovanje pokaže, da je razlog za to počasen SPI vmesnik. Edina možna rešitev te težave je zamenjava zaslona z drugim, ki namesto serijskega SPI vmesnika uporablja paralelni vmesnik. Ker počasen izris ne predstavlja hude težave se naposled odločimo za nadaljnjo uporabo obstoječega zaslona.

Sledi še povezovanje vhodne naprave - rotacijskega kodirnika, ki predstavlja glavno vhodno metodo. Kodirnik lahko neskončno obračamo levo ali desno ali nanj pritisnemo (deluje kot gumb). Te tri operacije so tako dovolj za izdelavo celotnega uporabniškega vmesnika. Uporabniške vmesnike z rotacijskimi kodirniki pogosto najdemo v različnih napravah. Pogosto se uporabljajo za kontrolo glasnosti na zvočni opremi namesto potenciometrov.

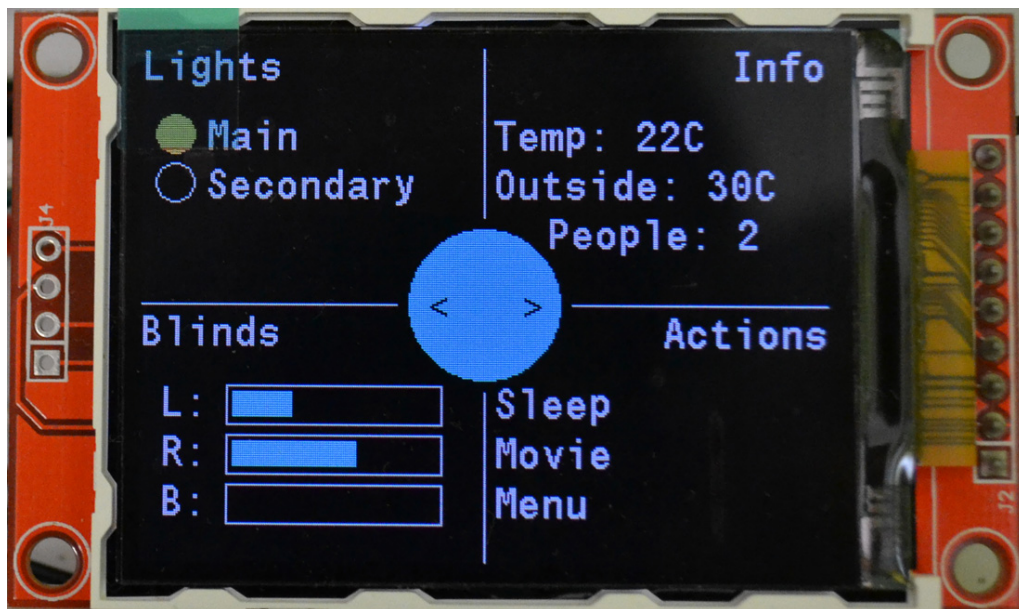
Rotacijski kodirnik preko dveh izhodnih kanalov povzroči ustrezne impulze, iz katerih lahko ugotovimo smer vrtenja. Uporabljeni mikrokrmilnik omogoča strojno branje rotacijskih kodirnikov, kar je bilo uporabljeno v tej nalogi. Strojno branje kodirnikov je izvedeno preko časovnikov. Namesto periodične ure v časovnik dovedemo impulze kodirnika. Ob obratu kodirnika se vrednost časovnika poveča ali zmanjša za število obratov. Iz tega lahko v programu sklepamo ali je uporabnik zavrtel gumb v levo ali desno ter za koliko ga je obrnil. Števec premikov hranimo v posebnem statičnem razredu, da lahko dostopamo do njega iz kjerkoli. Podobno velja tudi za osrednji gumb. Ob pritisku na gumb se sproži prekinitev, ki v statični razred postavi zastavico, da je bil gumb pritisnjen.

2.2.3 Uporabniški vmesnik

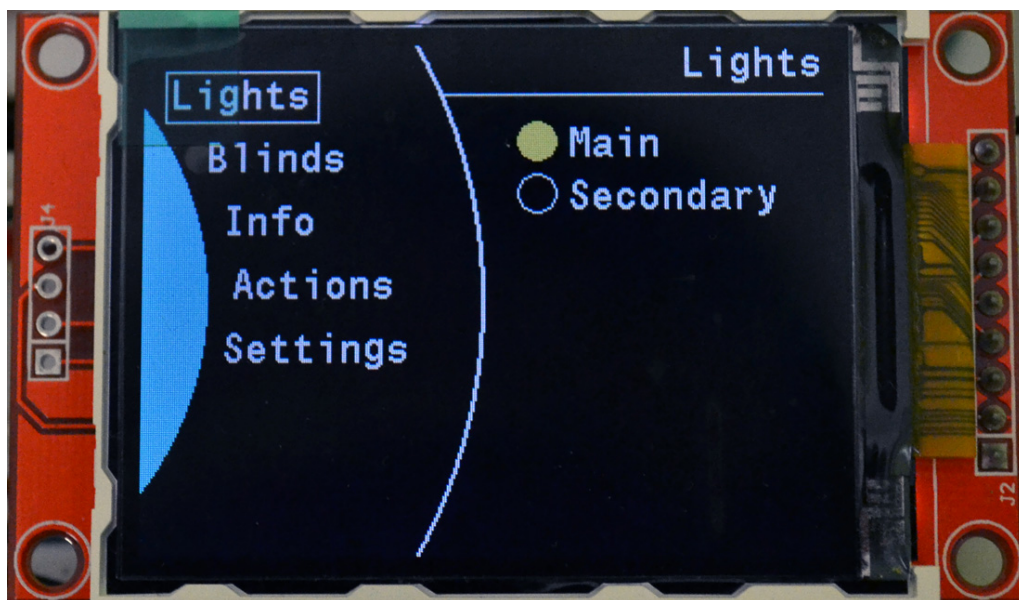
S tem je bila pripravljena osnova za izdelavo sistema. Pripravljena je bila osnovna vhodno/izhodna strojna oprema. Sledi načrtovanje uporabniškega vmesnika. Ta mora biti primeren za manjši zaslon in prilagojen upravljanju z rotacijskim kodirnikom.

Pred izdelavo prototipa uporabniškega vmesnika preverimo še ali obstaja kakšna knjižnica za izdelavo uporabniških vmesnikov. Hitro ugotovimo, da obstaja nekaj izbire na področju grafičnih knjižnic za vgrajene sisteme, vendar so te večinoma prilagojene za zelo specifične mikrokrmilnike in krmilnike zaslonov. Tako najdemo kar nekaj knjižnic vendar te ne delujejo z izbranim mikrokrmilnikom saj ta ne vsebuje grafičnega krmilnika, ki je prisoten na nekaterih drugih modelih iz iste serije. Po neuspešnem iskanju primerne knjižnice se odločimo za izdelavo lastne grafične knjižnice.

Sledi izdelava dveh prototipov grafičnega vmesnika. Prototipa služita le za prikaz izgleda in ne ponujata nobene funkcionalnosti. Prvi (prikazuje ga slika 2.4) skuša prikazati čimveč informacij na enem zaslonu kar omogoča hitrejši pregled nad vsemi podatki o sobi. To ga naredi manj preglednega in manj prijaznega uporabniku. Majhen zaslon pa omeji število podatkov, ki jih lahko na zaslonu hkrati prikažemo. Drugi (prikazuje ga slika 2.5) na-



Slika 2.4: Prvi prototip uporabniškega vmesnika



Slika 2.5: Drugi prototip uporabniškega vmesnika

mesto vseh podatkov prikazuje le en del podatkov hkrati. To ga naredi bolj preglednega in bolj uporabniku prijaznega vendar nekoliko zmanjša hitrost upravljanja saj mora uporabnik brskati po menijih preden lahko pride do nekaterih podatkov. Po nasvetu nekaj možnih uporabnikov in tehtnem premisleku za končni sistem izberemo drugi prototip.

Temu sledi začetek dela na lastni grafični knjižnici. Ta del razvoja je hkrati tudi tisti, ki je trajal najdlje. V tem delu so bili razviti nekateri osnovni gradniki grafičnega vmesnika. Sem spadajo menijski gumbi, indikatorji, kazalniki poteka (angl. *progress bar*) in drugi. Vsi so v osnovi preprosti in sestavljeni iz gradnikov, ki jih lahko izriše knjižnica za delo z zaslonom LCD (krog, kvadrat, besedilo, črte). Tako zmanjšamo odvisnost našega sistema od izbranega krmilnika zaslona. V primeru zamenjave zaslona moramo tako v teoriji popraviti le grafično knjižnico, da bo delovala z novim zaslonom. Na sliki 2.5 so na levi strani prikazani menijski gumbi (prvi izmed teh je izbran), na desni pa sta prikazana dva indikatorja. Ker je bil za razvoj uporabljen programski jezik C++ je bilo na tem mestu ustvarjeno več razredov, ki dedujejo lastnosti eden od drugega.

Menijski gumbi so vredni posebnega poudarka saj se mora ob pritisku nanje zgoditi določena akcija. Tukaj na pomoč priskočijo dodatki v standardu programskega jezika C++ različice 2011 (C++11). Sem spadajo Lambda funkcije, ki so v splošnem znane kot zaprtje (angl. *closure*) in funkcijske spremenljivke. Ti omogočajo, da lahko funkcijo (oziroma kazalec nanjo) shranimo v spremenljivko posebnega tipa. Kadar uporabnik pritisne na gumb mora program le poklicati funkcijo, ki je shranjena v spremenljivki tega gumba, ne da bi bilo potrebno vedeti karkoli več o tej funkciji. Lambda funkcije pa nam omogočijo, da lahko notranja funkcija dostopa tudi do spremenljivk zunaj njenega dosega (angl. *scope*).

Navigacija med meniji in podmeniji je izvedena s pomočjo preprostega sklada. Ob prehodu v podmeni se na sklad shranijo vsi grafični elementi in na zaslonu izrišejo novi. Ti lahko vodijo v naslednji podmeni in tako dalje. Ob vrnitvi iz podmenija se trenutni grafični elementi izbrišejo iz pomnilnika,

stari elementi pa se vzamejo iz vrha sklada in ponovno izrišejo. S tem lahko izvedemo poljubno konstrukcijo menijev poljubne globine. Sam sklad je izveden s pomočjo vektorja (*std::Vector*), ki je standardna podatkovna struktura vključena v programski jezik C++. Za menijski sistem uporabljamo dva glavna vektorja. Prvi hrani trenutno izrisane menijske elemente, drugi pa hrani vektorje menijskih elementov iz nadrejenih menijev.

Hitra predstavitev poteka prehoda iz glavnega menija v meni za upravljanje z lučmi:

1. Začne se izvajanje procesa za izris menija.
2. Proces preveri ali je prišlo do uporabniškega vhoda. Nato izračuna za koliko je bil zavrten enkoder in okoli novo izbranega menijskega elementa izriše okvir (ter ga izbriše iz prej izbrane opcije). Nato preverimo, ali je prišlo do pritiska na enkoder (potrditev).
3. Ker je prišlo do pritiska gumba za potrditev pomeni, da uporabnik želi vstopiti v izbran meni. Proces pogleda v vektor s trenutno prikazanimi menijskimi elementi in izvede funkcijo, ki je shranjena v posebni spremenljivki, ki je del elementa izrisanega na zaslon.
4. Funkcija izbriše potrebni del zaslona, ustvari nov prazen vektor in za vsako luč ustvari nov menijski element (indikator). Temu nastavimo ime in stanje. Za vsako luč program ustvari še lambda funkcijo, ki se bo izvedla v primeru, da uporabnik izbere ta element. Menijski element izrišemo in shranimo v novo ustvarjeni vektor. Funkcija nato stari vektor trenutno prikazanih elementov porine na sklad in vektor s trenutnim prikazanimi elementi zamenja z novo ustvarjenim vektorjem.
5. S tem je bil narejen prehod iz enega menija v nov meni. Funkcija se s tem zaključi.
6. Proces za izris menija se s tem za nekaj časa ustavi. Operacijski sistem nadaljuje z izvajanjem ostalih procesov. Po pretečenem času bo proces

za izrisovanje menija ponovno izveden. Izvajanje se bo ponovno začelo v točki 1.

Ta način izvajanja omogoča krmilniku, da porabi, kar se da malo časa za poganjanje uporabniškega vmesnika. Spremembe se zgodijo le, kadar pride do uporabnikovega vhoda. Ostal čas lahko sistem izvaja ostale procese ali čaka na vhod.

Med kasnejšim razvojem so bili razviti še dodatni spustni meniji in meniji za izbiranje ter vnosna polja za besedilo. Za vnašanje besedila je bila razvita tudi preprosta zaslonska tipkovnica. Med samim razvojem so bili po potrebi nekateri grafični elementi popravljeni in prilagojeni potrebam.

2.2.4 Nadzor luči

Ob koncu razvoja osnovnega uporabniškega vmesnika se je lahko začel razvoj dejanskega sistema za avtomatizacijo. Prvi del, ki je bil razvit je sistem za avtomatizacijo luči. Sistem je zasnovan, da lahko ločeno upravlja z največ petimi lučmi. Ta omejitev je bila vpeljana predvsem zaradi lažje grafične predstavitve in omejitev s pomnilnikom.

Iz stališča strojne opreme so bili tu dodani trije moduli za brezžično komunikacijo. To so radijski oddajnik (frekvence 433MHz), radijski sprejemnik in Bluetooth LE modul. Bluetooth LE modul je bil dodan kot možnost nadaljnjega razvoja, vendar nikoli uporabljen.

Kontrola luči je tako izvedena preko radijskih modulov frekvence 433MHz. To je frekvenca, ki je pogosto uporabljena v široko dostopnih brezžično vodenih stikalih. To nam prihrani razvoj ločene strojne opreme za nadzor luči in zmanjša ceno celotnega sistema. Brezžični moduli so tako kot brezžična stikala cenovno ugodni. Za upravljanje so bila izbrana brezžična stikala podjetja CoCo. Podjetje omenjeno že v uvodnem poglavju izdeluje modularne naprave za avtomatizacijo doma. Ponujajo različne oblike stikal in senzorjev. Avtorjeva soba je tako opremljena z enim stikalom za glavno luč in drugim zatemnilnim stikalom (angl. *dimmer*) za manjše halogenske luči. Identična



Slika 2.6: Brezžično stikalo ACM-100 proizvajalca CoCo [12]

(uporabljajo enak protokol) stikala lahko najdemo tudi pri množici drugih proizvajalcev. Nadvse primerna so zaradi enostavne vgradnje, saj jih moramo le vgraditi namesto obstoječega stikala oziroma v seriji z obstoječim stikalom. Ker so stikala brezžična, nam tako ni potrebno polagati nobenih dodatnih kablov. Primer takega stikala prikazuje slika 2.6. Kompatibilna bi morala biti vsa stikala podjetja CoCo ali Intertechno, ki za določanje naslova naprave uporabljajo mehanični izbirnik. Stikala ostalih proizvajalcev niso bila preizkušena.

Iz stališča programske opreme je bilo potrebno dodati izvorno kodo, ki bo pravemu stikalu poslala ustrezen signal preko brezžičnega oddajnika. Protokol uporabljen v teh stikalih uporablja enostavno naslavljanje. Naslov naprave je sestavljen iz dveh delov - črke od A do P, ki predstavlja skupino in številke, ki predstavlja napravo. Na brezžičnem stikalu s preprostim izbirnikom nastavimo ustrežno skupino in številko naprave. Za pošiljanje ustreznih signalov je bila uporabljena odprtokodna knjižnica (*433MHz for Arduino* [19]), ki podpira nekaj brezžičnih protokolov različnih proizvajalcev.

V okviru izdelave tega sistema je bila ta knjižnica prenesena iz platforme Arduino na platformo ARM. V sklopu prenosa je bilo potrebno napisati nekaj funkcij za delo s časovniki ter ustrezno zamenjati nekaj podatkovnih tipov. Ker knjižnica podpira tudi protokole drugih proizvajalcev je sistem še toliko bolj razširljiv.

2.2.5 Nadzor okenskih žaluzij

V večini sistemov za avtomatizacijo prostorov je nadzor okenskih žaluzij izveden z uporabo posebnih okenskih žaluzij, ki imajo vgrajene motorje za premikanje. Ker take žaluzije niso pogoste v domovih in stanejo kar nekaj denarja je bilo potrebno najti boljšo rešitev.

V sklopu prejšnjega sistema je bil nadzor žaluzij izveden s pomočjo navadnega enosmernega motorja in ločenega potenciometra. Tako lahko sistem dobi povratno informacijo o tem, kako je obrnjena žaluzija. Težava je v ločenem motorju in potenciometru. Do žaluzije je potrebno napeljati nekaj kablov in motor, ter potenciometer ločeno pritrditi na žaluzijo. Ker navadne okenske žaluzije niso namenjene vgrajevanju motorjev morata biti tako potenciometer kot motor nameščena zunaj ohišja žaluzij kar nekoliko pokvari izgled.

Za nov sistem je bila izbrana boljša rešitev. Namesto ločenega motorja in potenciometra je bil za obračanje izbran servo motor. Manjše servo motorje najdemo v svetu modelarstva za premik kril modelov letal, krmiljenje modelov avtomobilov in podobno. Prav tako so uporabljeni v veliko različnih projektih domače robotike. Odlikuje jih zelo nizka cena in enostavna dobavljivost. Kupimo jih namreč lahko v vsaki trgovini z modelarsko opremo. V enem ohišju vsebujejo preprost enosmerni motor, potenciometer in elektroniko za krmiljenje. Za delovanje potrebujejo le tri žice - napajanje in krmilni signal. Z ustreznim krmilnim signalom motorju povemo, na katero lokacijo naj premakne ročico. Ker ima vgrajen potenciometer lahko tako z zadovoljivo natančnostjo ugotovi na kateri lokaciji je ročica. Večina takih servo motorjev ima za potrebe krmiljenja žaluzij veliko pomanjkljivost. Zmožni so se obrniti

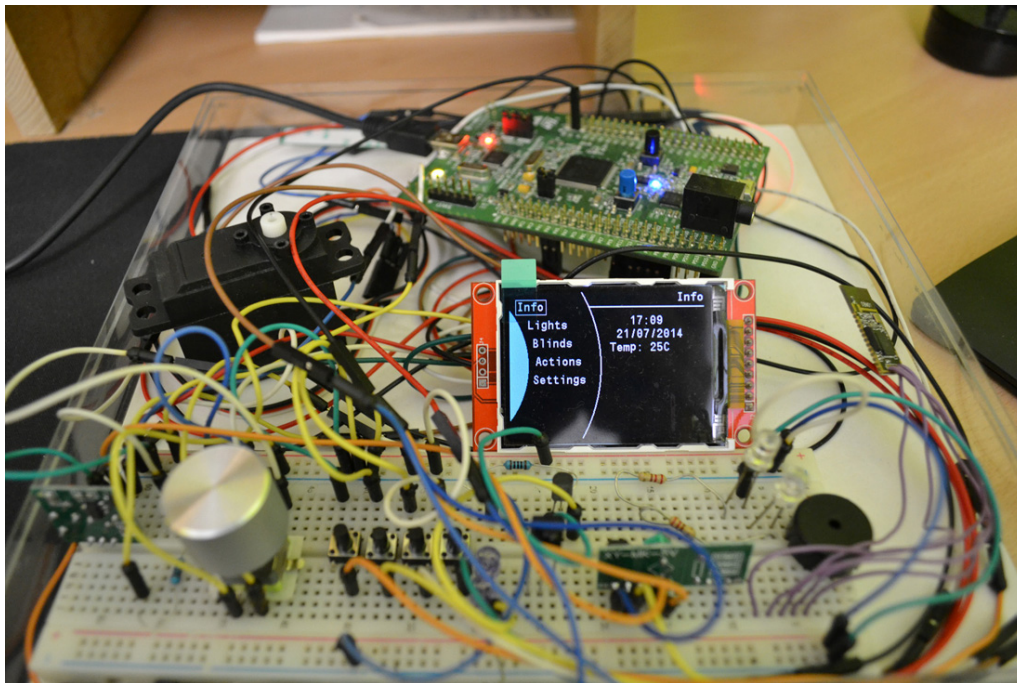


Slika 2.7: Motor SW4805-4PA proizvajalca HobbyKing [15]

za največ 180° (ročico lahko premaknemo za največ polovico kroga) kar jih naredi neprimerne za kontrolo žaluzij, ki potrebujejo nekoliko večji obrat, če želimo žaluzijo obrniti pod katerikoli kotom. Po raziskovanju ugotovimo da obstaja še ena možnost. V modelarstvu lahko najdemo tudi servo motorje namenjene modelom jadrnic (angl. *sailwinch servo*). Ti zaradi svoje izvedbe namesto polovičnega obrata omogočajo približno 5 obratov kar je dovolj za nadzor žaluzij. Primer takega motorja prikazuje slika 2.7. Uporabljen motor z oznako *SW4805-4PA* je bil kupljen preko spletne trgovine HobbyKing [15].

Vsi modelarski servo motorji so krmiljeni s pomočjo pulzno-širinske modulacije (angl. *PWM*). Pri pulzno-širinski modulaciji preko ene žice pošljamo preprost signal neke frekvence. PWM napravo (v tem primeru motor) nadzorujemo s spreminjanjem časa, v katerem je signal v logičnem stanju 1. Mikrokontrolniki za namen generiranja PWM signalov uporabljajo časovnike. Za nadzor žaluzij je bilo tako potrebno le nastaviti časovnik, naj deluje v načinu PWM in nastaviti želene dolžine signalov kar motor premakne na željeno mesto.

V nadaljevanju je bila dodana še možnost, da uporabnik nastavi tri stanja žaluzije - žaluzija obrnjena do konca navzdol, sredina in žaluzija obrnjena do



Slika 2.8: Prototip sistema med razvojem

konca navzgor. To prepreči, da bi uporabnik poškodoval žaluzijo ali motor, kadar bi hotel obrniti žaluzijo preko dovoljenih mej. Dodamo še možnost, da uporabnik s pritiskom na tipko izbira med temi tremi stanji žaluzije.

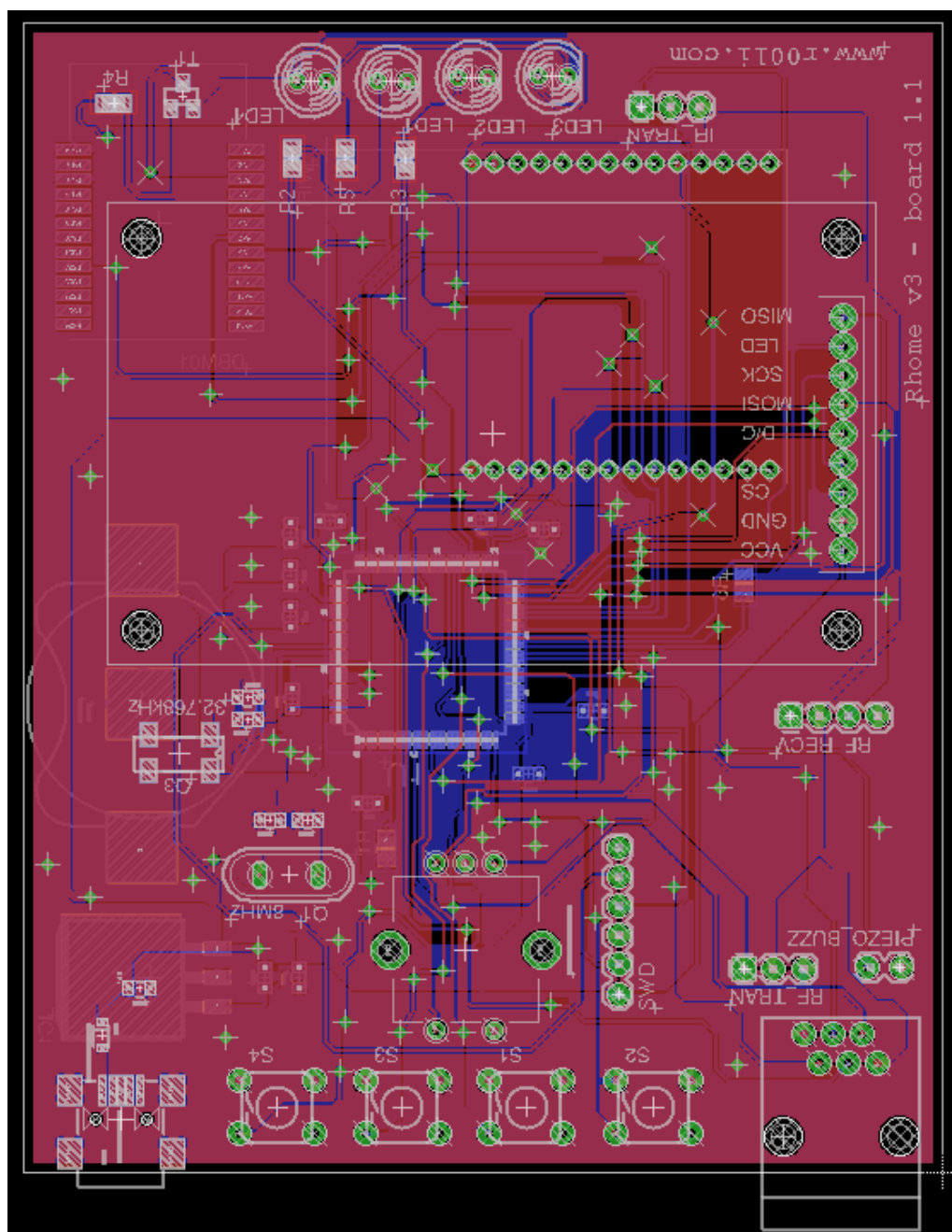
2.2.6 Vključitev dodatne strojne opreme

Naslednji korak v razvoju je bilo dodajanje vse potrebne strojne opreme. To je bilo potrebno storiti, preden smo lahko izrisali tiskano vezje za sistem. Dodan je bil modul za brezžično povezovanje (več o tem v nadaljevanju). Dodane so bile še štiri tipke za hitrejše upravljanje sistema. Prav tako so bili dodani IR sprejemnik in IR LED diode kot možnost nadaljnjega razvoja. Za opozarjanje uporabnika sta bili dodani še dve barvni LED diodi. To je omogočilo testiranje strojne opreme pred izdelavo tiskanega vezja. Slika 2.8 prikazuje prototip sistema med delovanjem s priključeno vso strojno opremo.

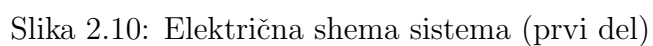
2.2.7 Načrtovanje in izdelava tiskanega vezja

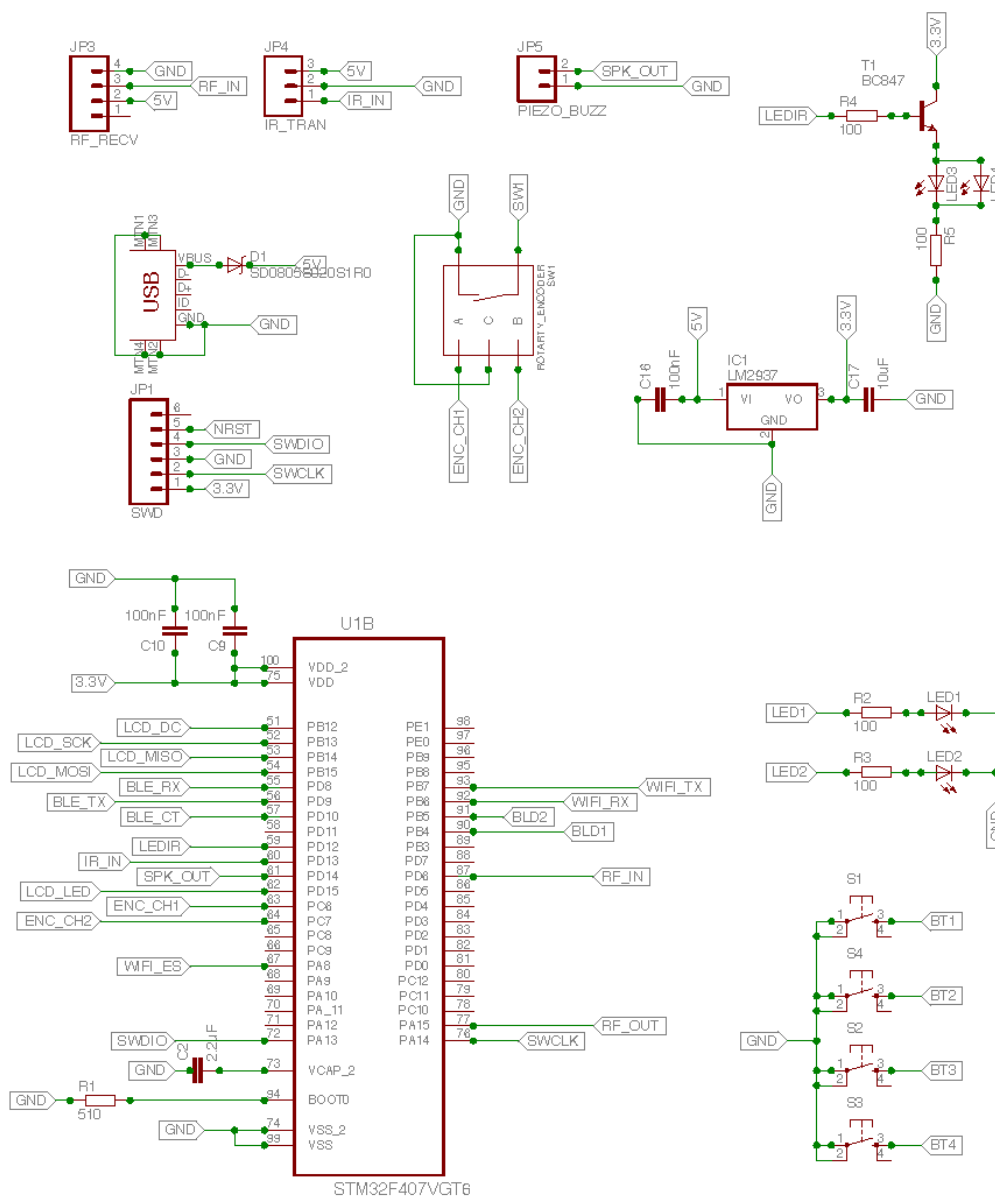
Temu je sledila izdelava tiskanega vezja za sistem. Izbrani mikrokrmilnik že vsebuje nekaj nujno potrebne strojne opreme (oscilatorji, enota za napajanje, ...) za delovanje, kar močno poenostavi načrtovanje. Poleg mikrokrmilnika za osnovno delovanje potrebujemo le nekaj kondenzatorjev in en upor. Zaradi večje natančnosti pri merjenju časa je bil dodan še zunanji kristal. Za potrebe ure realnega časa je bil dodan še urin kristal in držalo za gumb baterijo. Za napajanje celotnega sistema je bil dodan standarden mini-USB priključek. Ta neposredno napaja motorje za premik žaluzij, 433MHz radijske module in WiFi modul. Ti vsi delujejo na napetosti 5V. Mikrokrmilnik in ostale naprave za delovanje potrebujejo 3.3V. Zaradi tega je bil dodan še regulator napetosti. Dodan je bil še priključek RJ-11 za povezavo z motorji za krmiljenje žaluzij. S programom Eagle je bila najprej narisana električna shema celotnega sistema. V veliko pomoč pri risanju sheme je bila dokumentacija podjetja STM, kjer je natančno opisano kakšne komponente moramo uporabiti pri načrtovanju. V še dodatno pomoč je bila tudi shema uporabljene razvojne ploščice. Shemo celotnega sistema prikazujeta slika 2.10 in slika 2.11.

Ko je bila shema končana je sledil izris dejanskega tiskanega vezja. Vse izbrane komponente je bilo potrebno ustrezno razporediti po čim manjši površini in jih povezati glede na prej narisano shemo. Pri razporejanju komponent so bile vse komponente najprej postavljene na približno mesto kamor sodijo. Priključki se morajo zaradi svoje oblike namreč nahajati na robu vezja. Zaslon sodi v osrednji prostor (predstavlja namreč glavni del uporabniškega vmesnika) pod njim pa sodijo vhodne komponente. Prav tako si želimo vezje narediti čim tanjše zaradi lažje vgradnje v morebitno ohišje. Sledi bolj natančno razporejanje komponent. Pri tem je potrebno upoštevati različne velikosti in oblike komponent in njihove električne zahteve. Kondenzatorji, ki gladijo vhodno napetost mikrokrmilnika se morajo namreč nahajati čim bližje samemu čipu (kot je opisano v navodilih proizvajalca). Podobno velja tudi za oba kristala. Vse komponente, ki skrbijo za napajanje (regulator napetosti, baterija in ostale) skušamo združiti v en rob

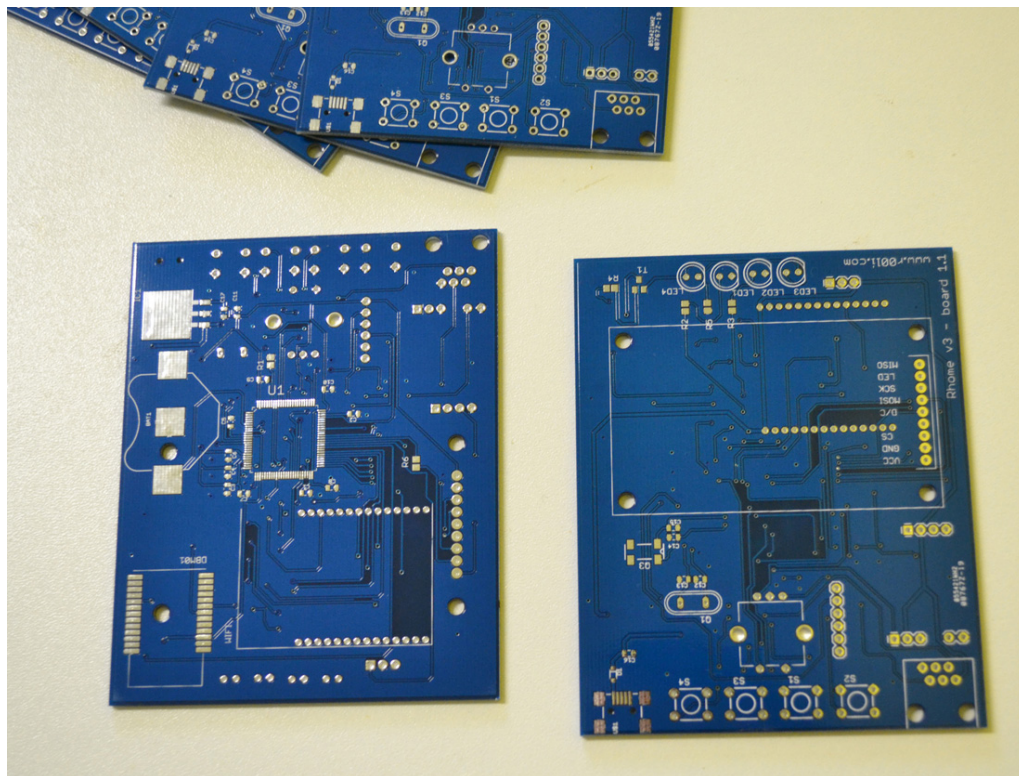


Slika 2.9: Izrisano tiskano vezje pripravljeno za izdelavo





Slika 2.11: Električna shema sistema (drugi del)



Slika 2.12: Izdelana tiskana vezja

tiskanega vezja. Ko so komponente ustrezno razporejene jih lahko povežemo med seboj. Pri tem kako komponento tudi nekoliko prestavimo, da lažje narišemo povezave. Pri povezovanju nas nekoliko omejuje tudi sama velikost tiskanega vezja. Končno tiskano vezje na sliki 2.9 je tako veliko 10x8cm, kar je tudi največ kar lahko narišemo z brezplačno različico programa EAGLE. V primeru uporabe dodanih nivojev (angl. *layer*) bi lahko močno prihranili na velikosti samega vezja. To bi sicer pomenilo, da bi morali uporabiti katero od plačljivih različic programa EAGLE in dražjo izdelavo samih tiskanih vezij.

Ko je bil izris tiskanega vezja končan je sledila njegova izdelava. Za izdelavo je bilo uporabljeno Kitajsko podjetje Elecrow [17]. To podjetje je bilo izbrano zaradi zelo ugodne cene za komplet sedmih izdelanih vezij. Ta so bila izdelana in dostavljena v Slovenijo v roku enega tedna. Slika 2.12

prikazuje izdelana tiskana vezja pripravljena za uporabo.

Sledilo je še sestavljanje vezja. Večina elektronskih komponent je bila kupljena pri Britanskem podjetju Farnell [14]. Ostale komponente so bile kupljene pri večih prodajalcih iz spletne strani Ebay.

2.2.8 Shranjevanje podatkov

Vse do tega trenutka so bili vsi podatki za krmiljenje luči (naslovi stikal) in krmiljenje žaluzij zapisani v sami programski kodi sistema. Ker je bil sistem načrtovan s prilagodljivostjo v mislih to ni bila zadovoljiva rešitev. Zaradi tega je bilo potrebno najti način za dodajanje in brisanje nastavitev med delovanjem. Ker se RAM ob izgubi napajanja izbriše to ni bila ustrezna rešitev. Uporabljeni mikrokrmilnik ponuja dve možnosti za trajno shranjevanje podatkov. Ena možnost je uporaba varnostnega pomnilnika SRAM (angl. *backup SRAM*). To je posebni del pomnilnika v katerega lahko ob izgubi napajanja prepisemo vsebino, ki jo želimo shraniti. Podatki v tem delu pomnilnika ostanejo vse dokler je prisotno baterijsko napajanje. Gumb baterija, ki se uporablja za poganjanje ure realnega časa lahko tako napaja ta del pomnilnika in prepreči njegov izbris. Pričakovana življenjska doba ene baterije je približno dve leti.

Druga možnost (ki je bila izbrana kot končna rešitev) je uporaba vgrajenega pomnilnika flash za shranjevanje podatkov. Mikrokrmilnik nam omogoča, da del pomnilnika namenjenega shranjevanju programske kode rezerviramo za shranjevanje podatkov. Težava je le v načinu delovanja pomnilnika flash. Preden lahko vanj podatke vpišemo mora biti pomnilnik izbrisan. Pri pomnilniku flash lahko izbrišemo le celoten sektor in ne posameznega bita. Pomnilnik tega mikrokrmilnika je razdeljen na 11 sektorjev. Za shranjevanje je bil izbran zadnji sektor (ki je hkrati tudi največji in obsega 128kB).

Shranjevanje je izvedeno na preprost način. Ob zagonu sistem prebere vse podatke iz pomnilnika flash v RAM. Branje je izvedeno na preprost način. Za vsak del, ki ga želimo shraniti je rezervirana zadostna količina prostora. Tako lahko shranimo 5 različnih luči, 4 različne žaluzije, 50 gumbov daljinskega

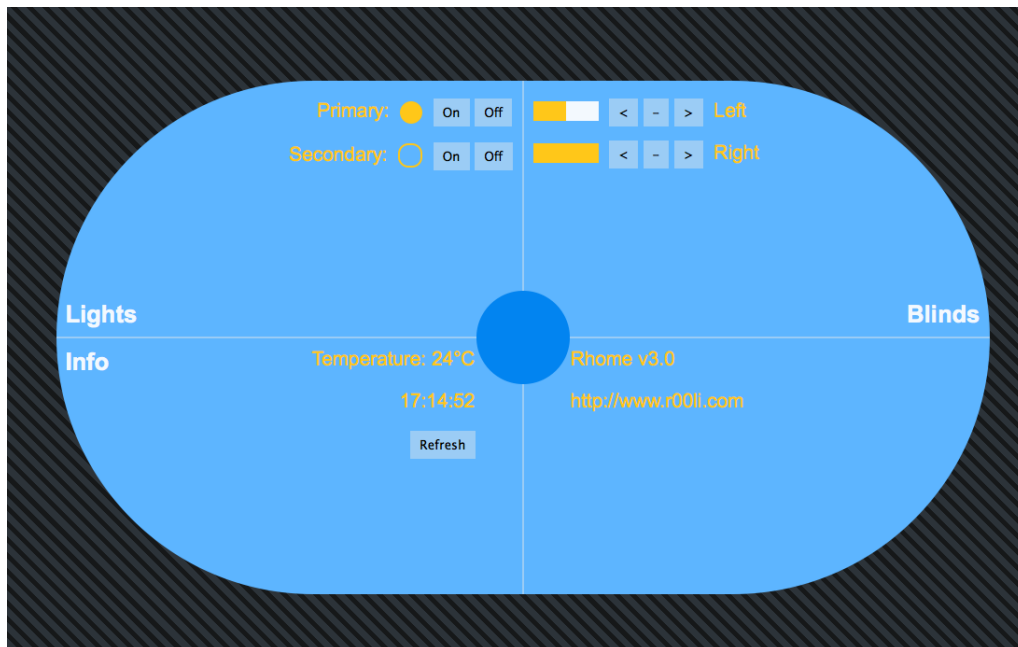
upravljalnika in še nekaj dodatnih podatkov. Ob zagonu sistem prebere podatke iz pomnilnika na točno določenem naslovu. Če je na pravem naslovu prisotna zastavica, da je podatek prisoten se ta podatek prebere, drugače se branje nadaljuje na naslednjem naslovu vse dokler ne pridemo do konca možnega obsega podatkov.

Sistem od tu naprej pomnilnika flash ne bere več. Kadar, uporabnik spremeni kako od nastavitev se ta najprej popravi v RAMu, nato se pokliče funkcija za shranjevanje v pomnilnik flash. Ta funkcija najprej izbriše celoten sektor pomnilnika in nato vse podatke, ki jih je potrebno trajno hraniti prepíše iz RAMa nazaj v flash. Tako imamo v RAMu in v pomnilniku flash vedno shranjene pravilne podatke. S tem je bila potreba po nastavitvah sobe shranjenih v programski kodi odpravljena. Sektor 11, ki se uporablja za shranjevanje pa bi moral zadostovati za hranjenje vseh potrebnih podatkov o sobi in ponuja še dovolj rezervnega prostora za nadaljnjo širitev.

2.2.9 Spletni strežnik in oddaljeno upravljanje

Ena izmed pomembnih funkcij sistema je oddaljeno upravljanje, ki omogoča nadzor nad lučmi ali ostalimi napravami, kadar nas ni v prostoru. Druga uporabna funkcija je tudi možnost nadzora preko ostalih naprav (mobilni telefoni, tablice, ...) kadar smo v sobi. Tako lahko na primer med gledanjem televizije nastavimo žaluzije, da preprečimo bleščanje sonca.

Ker smo izbrali modul za brezžično povezovanje, ki podpira nižje nivojske protokole (TCP in UDP) nam tako ni potrebno vključevati nižje nivojske kode za povezovanje z internetom. Modul lahko nastavimo da deluje kot TCP/UDP strežnik ali odjemalec. Vse te parametre lahko ročno nastavimo preko modulovega spletnega vmesnika ali preko mikrokrmilnika s serijskimi ukazi. Hkrati izbrani modul ponuja istočasno delovanje kot strežnik in odjemalec (oziroma kot dva odjemalca ali kot dva strežnika) hkrati. Za tako delovanje potrebujemo dve serijski povezavi med modulom in mikrokrmilnikom. Ena povezava skrbi za eno povezavo medtem, ko druga skrbi za drugo. Ta način je bil izbran v našem sistemu. Tako lahko naš sistem hkrati deluje



Slika 2.13: Spletni vmesnik za upravljanje

kot spletni strežnik in nemoteno komunicira s kako drugo spletno storitvijo kot odjemalec.

Zaradi prijaznosti uporabniku je v sistem vključeno tudi preprosto nastavitevno okno za brezžično povezovanje. Uporabnik mora le vpisati ime dostopne točke, izbrati način varovanja povezave in vnesti ključ. V primeru, ko želi uporabnik spremeniti kako drugo nastavev povezave pa lahko to stori ročno preko spletnega vmesnika, ki je vključen v sam brezžični modul. To lahko stori tako, da se poveže na brezžični modul z neko drugo napravo (modul namreč lahko deluje kot dostopna točka) in obišče konfiguracijsko spletno stran.

Za potrebe oddaljenega upravljanja je bilo potrebno izdelati tudi preprost spletni strežnik. Ker sam brezžični modul poskrbi za podrobnosti protokola TCP dobimo od modula ustrezno besedilo, ki predstavlja zahtevek HTTP (kadar nekdo v spletni brskalnik vnese IP naslov našega sistema). Preprosti spletni strežnik, ki smo ga razvili nato odgovori s primernim odgovorom (kot

je zapisano v protokolu HTTP) in vrne preprosto spletno stran za upravljanje. V primeru, da sta bila v zahtevku HTTP prisotna pravilno uporabniško ime in geslo bo strežnik prikazal spletno stran za upravljanje (prikazuje jo slika 2.13), drugače bo prikazal stran za prijavo.

Prijava poteka na preprost način. Ko uporabnik vnese uporabniško ime in geslo se ta dodata v zahtevek HTTP, ta pa se ponovno pošlje. Težava take rešitve je v tem, da se uporabniško ime in geslo vedno pošiljata kot navadno besedilo kar predstavlja varnostno ranljivost.

Podobno se izvajajo tudi vsi ukazi uporabnika. Ob pritisku na gumb se ustrezni parametri dodajo zahtevku HTTP, ki se ponovno pošlje. Primer nekega zahtevka, ko uporabnik ni prijavljen izgleda kot (predpostavimo, da je IP naslov našega sistema 192.168.1.2 in spletni strežnik teče na vratih 8080):

192.168.1.2:8080

Ko uporabnik vnese uporabniško ime in geslo bo poslan zahtevek izgledal kot:

192.168.1.2:8080/up_ime/geslo

Ko pa na spletnem vmesniku izbere neko akcijo (na primer prižiganje prve luči) pa bo zahtevek izgledal kot:

192.168.1.2:8080/up_ime/geslo/lght/0/on

Ta način je bil izbran zaradi preproste izdelave saj je izdelava bolj naprednega spletnega strežnika lahko zelo zahtevna in časovno potratna, hkrati pa ne prinaša bistvenih izboljšav.

2.2.10 Daljinski upravljalnik in gumbi

Še ena izmed zelenih funkcij sistema je upravljanje preko daljinskega upravljalnika oziroma štirih dodatnih gumbov prisotnih na samem tiskanem vezju



Slika 2.14: Daljinski upravljalnik ITS-150 proizvajalca Intertechno [18]

sistema. Trenutno sistem omogoča upravljanje z daljinskimi upravljalniki, ki pošiljajo signale frekvence 433MHz. Primer tega je upravljalnik namenjen upravljanju brezžičnih stikal podjetja Intertechno, ki ga prikazuje slika 2.14. Identične daljinske upravljalnike najdemo tudi pri proizvajalcu CoCo.

Kaj določen gumb naredi ob pritisku lahko določi in spremeni uporabnik. V nastavitvah sistema lahko na primer izbere določeno luč (ali kako drugo napravo) in izbere, da želi za prižiganje ali ugašanje uporabiti daljinski upravljalnik. Sistem uporabnika nato pozove naj pritisne zeleno tipko na daljinskem upravljalniku, ki jo želi uporabiti. Ko uporabnik to stori se izračuna preprosta zgoščevalna vrednost (angl. *hash*) izbrane naprave in shrani skupaj z 32-bitno kodo gumba in zeleno akcijo (prižiganje ali ugašanje luči, odpiranje žaluzij, ...) v pomnilnik flash.

Ko uporabnik pritisne gumb na daljinskem upravljalniku sistem preveri vse shranjene kode gumbov. Če najde zadetek preveri še ali se shranjena zgoščevalna vrednost naprave ujema s kako trenutno prisotno napravo. V

primeru, da se, izvede akcijo, ki je bila shranjena.

Zaradi lažje izvedbe se popolnoma enako obnašajo tudi tipke na samem tiskanem vezju sistema. Tudi te se namreč obnašajo popolnoma enako kot daljinski upravljalnik. Uporabnik lahko v nastavitvah sistema vse akcije tudi izbriše in pregleda, kar mu omogoča spreminjanje funkcij tipk na daljinskem upravljalniku.

2.3 Predstavitev izdelanega sistema

V tem delu sledi še predstavitev sistema in izdelanega uporabniškega vmesnika. Slika 2.15 prikazuje izdelan sistem, ki prikazuje osnovni zaslon.

Slika 2.16 prikazuje osnovni zaslon sistema. Na osnovnem zaslonu imamo pregled nad trenutnim datumom in časom, temperaturo v sobi in stanjem luči ter žaluzij.

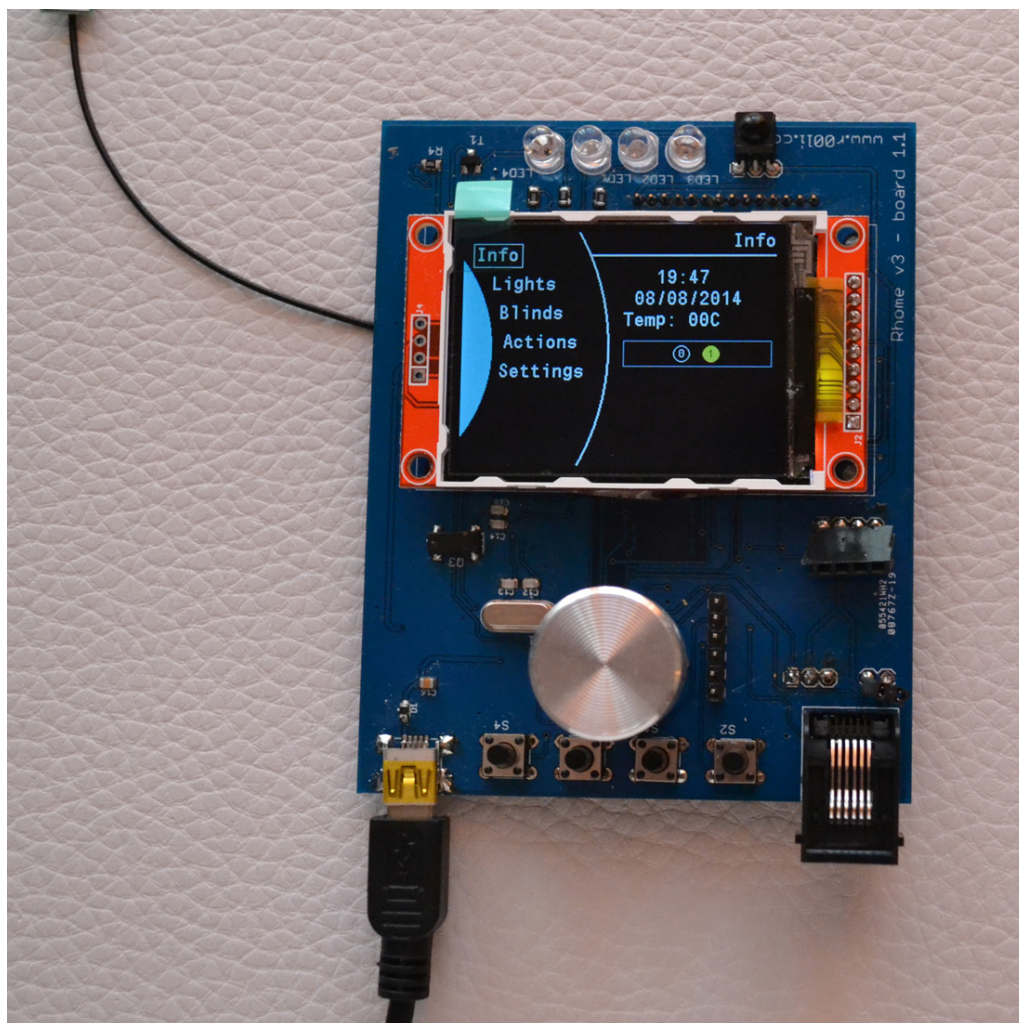
Slika 2.17 prikazuje seznam vseh luči v sobi. Iz tega seznama lahko izberemo zeleno luč ter jo prižgemo ali ugasnemo kar prikazuje slika 2.18. Prav tako je podprto zatemnjevanje luči (v primeru, da uporabljeno brezžično stikalo to podpira) kar lahko dosežemo z dvakratnim pritiskom gumba za vklop.

Slika 2.19 prikazuje meni z avtomatičnimi akcijami. Tako lahko izberemo spalni način, kar po naprej nastavljenem času avtomatično ugasne vse luči v sobi, in tako uporabniku omogoči, da odide v posteljo s prižganimi lučmi. Hkrati lahko vklopimo in izklopimo tudi avtomatično odpiranje in zapiranje žaluzij. Sistem lahko tako ob naprej določenem času zapre, ali odpre žaluzije.

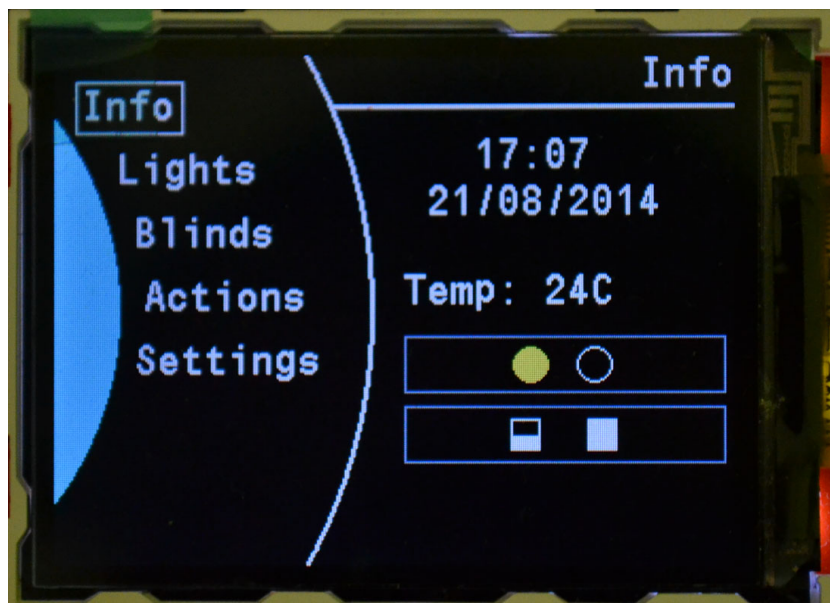
Slika 2.20 prikazuje seznam vseh žaluzij in njihov trenutni kot. Trenutno izbrano žaluzijo (v tem primeru leva žaluzija - kar označuje rumeno pobarvan prikaz stanja) lahko odpremo ali zapremo z vrtenjem glavnega gumba. Pritisk na glavni gumb spreminjanje zaključí.

Slika 2.21 prikazuje meni z nastavitvami kjer lahko sistem prilagodimo po svojih željah.

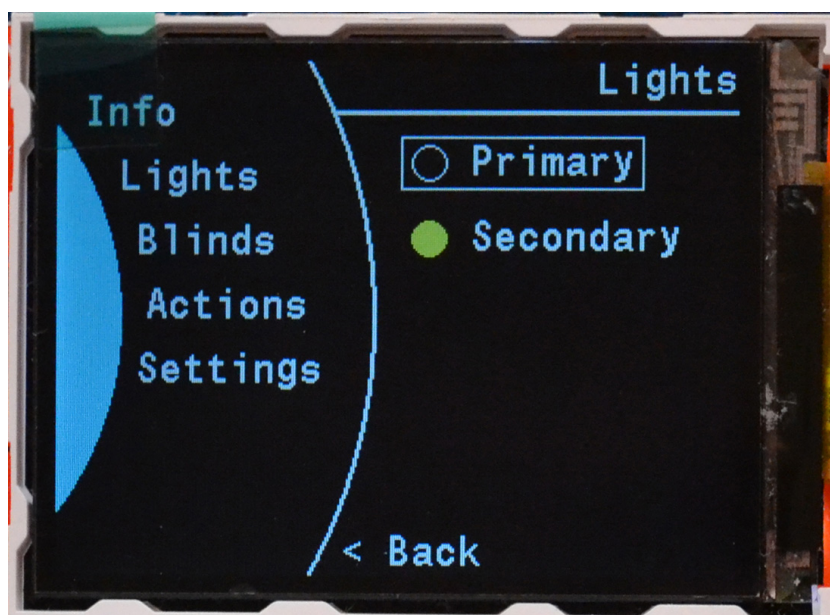
Slika 2.22 prikazuje meni z nastavitvami luči. Tukaj lahko dodamo novo



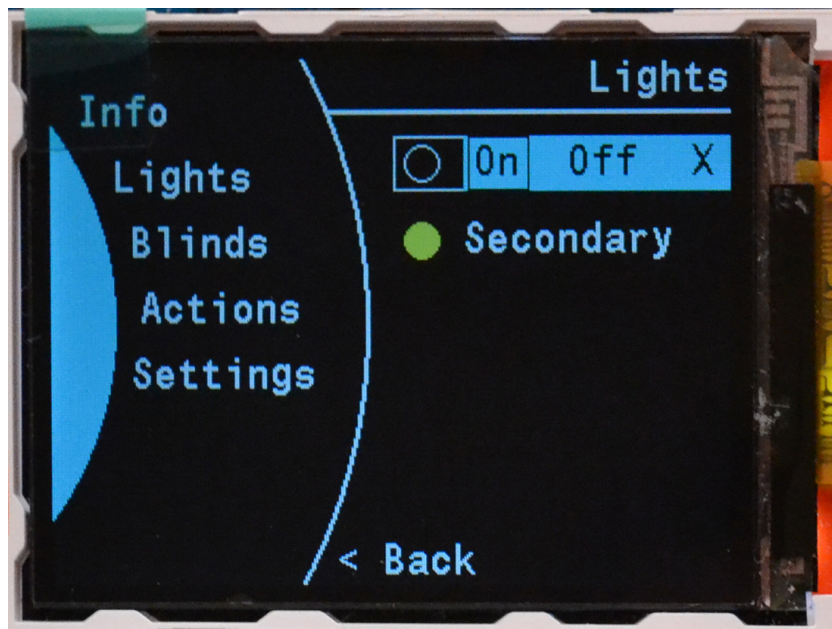
Slika 2.15: Izdelan sistem, ki prikazuje osnovni zaslon



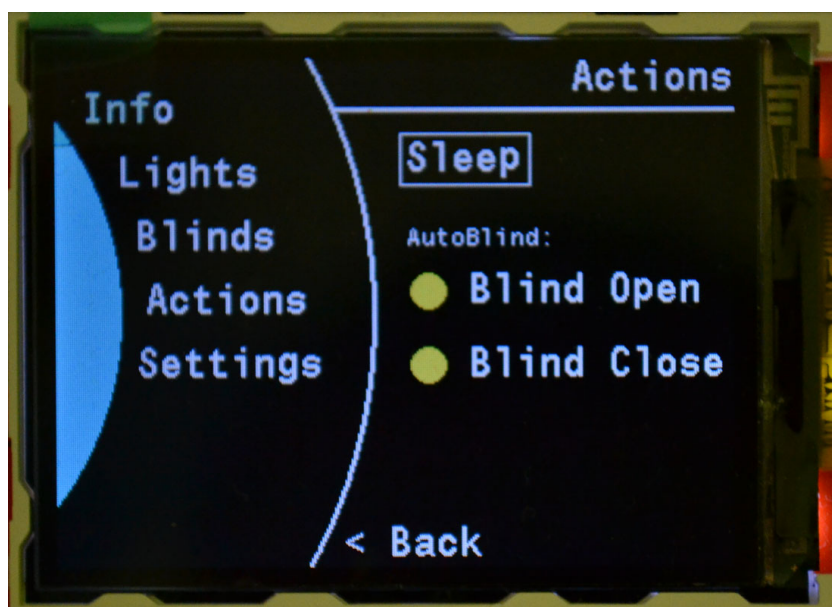
Slika 2.16: Osnovni zaslon sistema



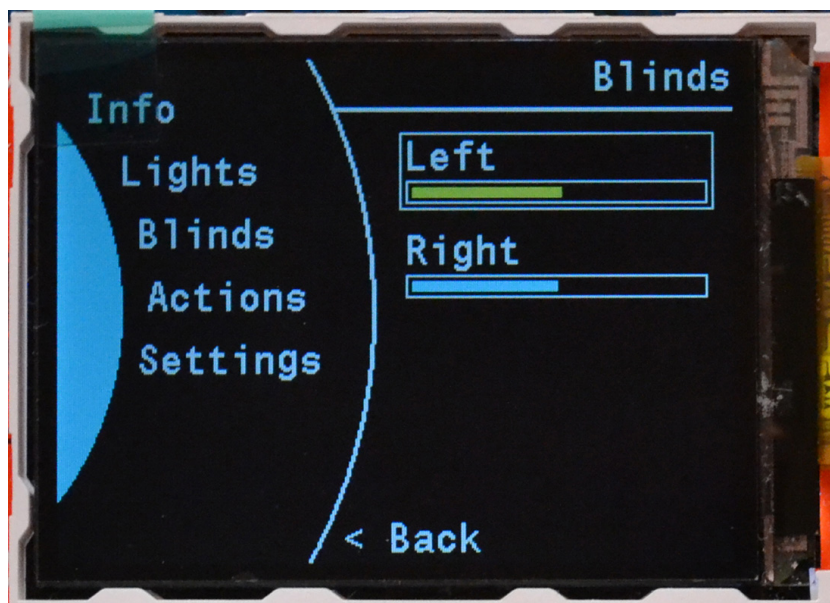
Slika 2.17: Zaslon, ki prikazuje stanje vseh luči v sobi



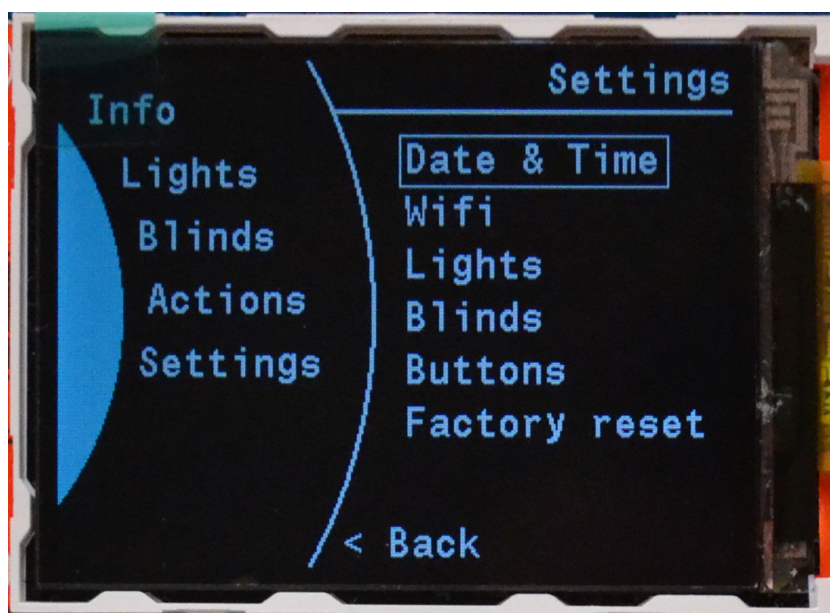
Slika 2.18: Prižiganje ali ugašanje luči



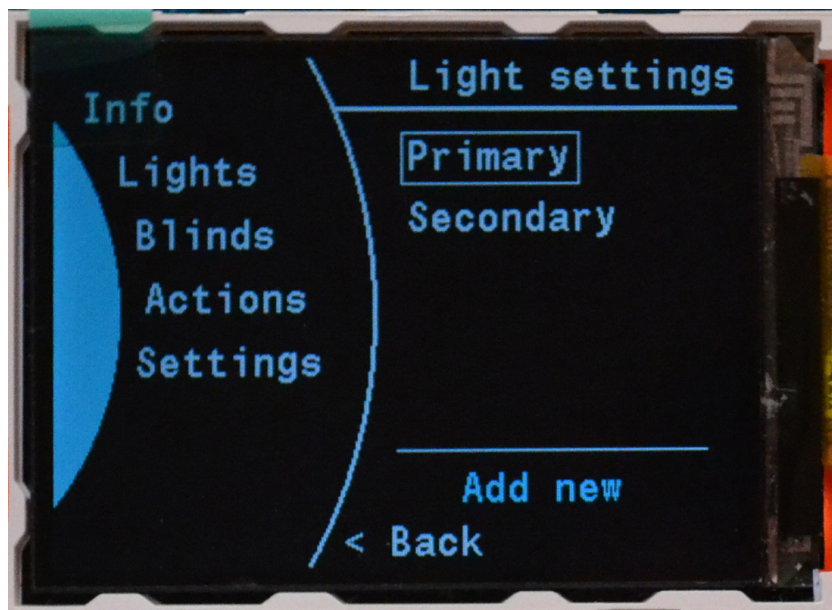
Slika 2.19: Meni z avtomatičnimi akcijami



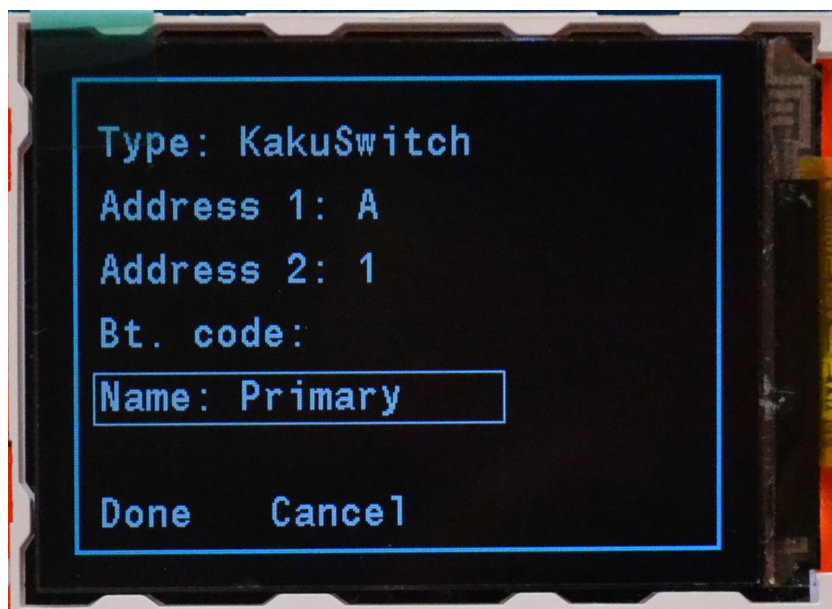
Slika 2.20: Seznam okenskih žaluzij



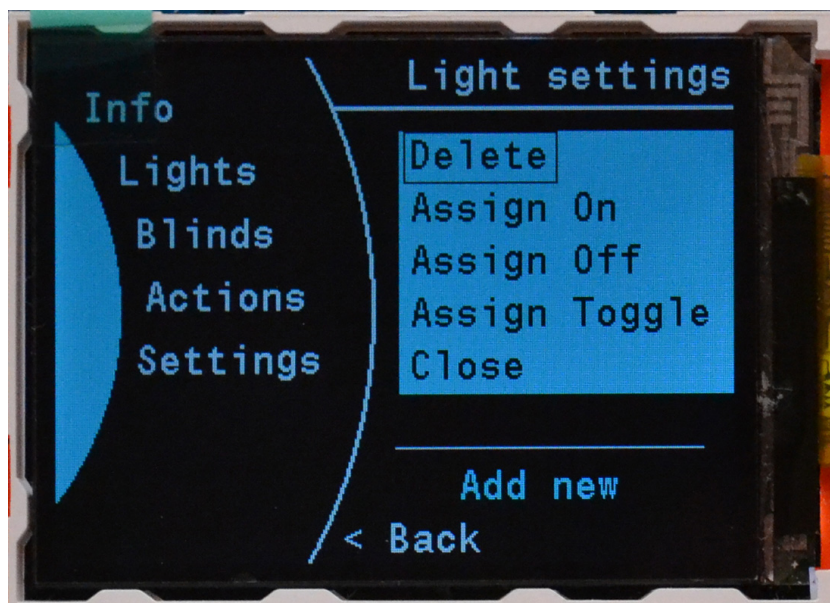
Slika 2.21: Meni z nastavitvami



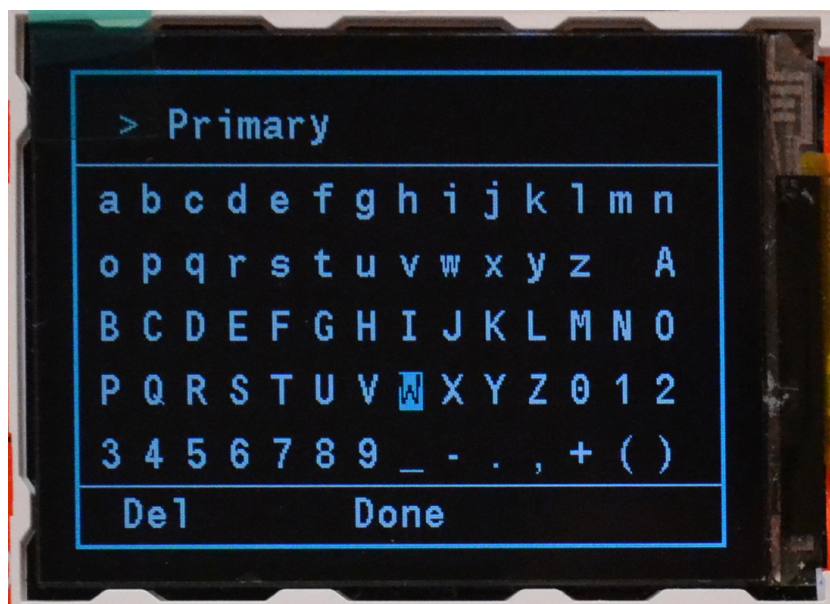
Slika 2.22: Meni z nastavitvami luči



Slika 2.23: Dodajanje nove luči



Slika 2.24: Urejanje obstoječe luči



Slika 2.25: Zaslonska tipkovnica

luč, kar prikazuje slika 2.23. Prav tako lahko katero od luči odstranimo ali ji določimo gumb za vklop ali izklop na daljinskem upravljalniku kar prikazuje slika 2.24.

Slika 2.25 prikazuje zaslonsko tipkovnico preko katere lahko vnašamo besedilo.

Poglavje 3

Sklepne ugotovitve

V sklopu te diplomske naloge je bil ustvarjen celoten sistem za avtomatizacijo sobe. Ta omogoča nadzor nad najpomembnejšimi vidiki bivanja v prostoru. V osnovi nam omogoča lokalno upravljanje z lučmi in okenskimi žaluzijami. Kasneje je bila v sistem vključena še možnost oddaljenega upravljanja preko spleta. To uporabnikom omogoča nadzor prostorov, tudi kadar jih ni doma, ali se nahajajo v kakem drugem prostoru. Vključena je tudi možnost lokalnega upravljanja preko daljinskega upravljalnika, kar močno poveča udobje pri uporabi sistema. Na prvi pogled tak sistem izgleda zelo preprost, vendar hitro ugotovimo, da je resničnost nekoliko drugačna.

Razvoj sistema je trajal približno šest mesecev. Od tega je najdlje trajal razvoj grafičnega uporabniškega vmesnika. Tu je bilo veliko večino časa porabljenega za izdelavo lastne grafične knjižnice. Uporaba kakega drugega mikrokrmilnika z eno izmed obstoječih grafičnih knjižnic bi močno skrajšala čas razvoja. V primeru, da bi bilo za razvoj na voljo nekoliko več časa bi sistem vključeval tudi upravljanje z multimedijskim predvajalnikom. Ker je razvoj grafičnega uporabniškega vmesnika trajal predolgo ta funkcionalnost v končni različici sistema manjka. Kar nekaj časa je bilo porabljenega tudi za izdelavo lastnega kompleksnega tiskanega vezja. Za ustrezno izbiro, postavitev in povezovanje komponent je namreč potrebnega kar nekaj časa in znanja. Kljub daljšemu času razvoja in težavam pri izvedbi uporabniškega

vmesnika pa končni sistem izpolnjuje vse zadane cilje.

Glavni cilj te naloge je bil ustvariti sistem za avtomatizacijo prostorov, s kar največ možnimi funkcijami in veliko možnostjo razširitve. Vse to za najnižjo možno ceno. Cena vse potrebne strojne opreme (vključno z dostavo) za sistem tako znaša približno 65EUR. V to ceno so vključene vse komponente za sistem, izdelava tiskanega vezja in dva motorja za obračanje žaluzij. Vse kar potrebujemo poleg tega je brezžično stikalo za vsako luč, ki jo želimo upravljati in preprost napajalnik. Če tudi to vključimo v ceno potrebne strojne opreme, lahko vsako posamezno sobo hiše ali stanovanja avtomatiziramo za manj kot 100EUR. To predstavlja privlačno konkurenčno možnost, še posebno v primerjavi z nekaterimi drugimi trenutno dobavljivimi sistemi, ki ponujajo podobno funkcionalnost.

Možnosti za nadaljnji razvoj sistema je veliko. Modularna zasnova z veliko različnimi možnostmi komunikacije ponuja še veliko prostora za širitev. Dostop do interneta lahko izkoristimo za povezovanje s številnimi različnimi spletnimi storitvami kar odpre številne možnosti. Sem spadajo prikazovanje elektronskih sporočil, opozorila iz socialnih omrežij in še veliko več. IR sprejemnik in IR diode lahko uporabimo za krmiljenje skoraj vseh multimedijskih naprav. Še ena možnost je ugašanje luči ali zapiranje žaluzij, kadar uporabnik prižge televizijski sprejemnik. Bluetooth LE modul lahko uporabimo za sledenje uporabnikov v stanovanju ali hiši. To nam omogoča avtomatično ugašanje luči, ko uporabnik odide iz sobe. Tukaj našete razširitve so popolnoma mogoče s trenutno uporabljeno strojno opremo in potrebujejo le nadgradnjo programske opreme. Z nekaj dodatne opreme in manjšo spremembo strojne opreme pa lahko uporabnost sistema še razširimo.

Ta diplomska naloga lahko služi kot dokaz, da se področje avtomatizacije prostorov širi iz sveta znanstvene fantastike v vsak dom. V naslednjih letih lahko verjetno pričakujemo velik razcvet pametnih naprav, ki se bodo lahko povezale v en centralni sistem. Pot do tja je kratka, še posebno v primeru, da bodo različni proizvajalci začeli uporabljati enake standarde. Z bolj množično uporabo podobnih sistemov pa lahko pričakujemo tudi močan spust cen.

Vsa izvorna koda, shema strojne opreme in ostale pripadajoče
informacije so na voljo na:

<https://github.com/r00li/RhomeV3>

Literatura

- [1] Home automation, dostopno na:
http://en.wikipedia.org/wiki/Home_automation.
- [2] What is KNX, dostopno na:
<http://www.knx.org/knx-en/knx/association/what-is-knx/index.php>.
- [3] Building automation, dostopno na:
http://en.wikipedia.org/wiki/Building_automation.
- [4] Arduino, dostopno na:
<http://www.arduino.cc/en/Guide/Introduction>.
- [5] ARM Cortex M, dostopno na:
<http://www.arm.com/products/processors/cortex-m/>.
- [6] Understanding Building Automation and Control Systems, dostopno na:
http://www.kmcecontrols.com/products/Understanding_Building_Automation_and_Control_Systems.aspx.
- [7] 1.5 Million home automation systems installed in the US, dostopno na:
<https://www.abiresearch.com/press/15-million-home-automation-systems-installed-in-th>.
- [8] Nest, dostopno na:
<https://nest.com>.

- [9] Phillips Hue, dostopno na:
<http://meethue.com/>.
- [10] X10, dostopno na:
<http://www.x10.com/>.
- [11] Insteon, dostopno na:
<http://www.insteon.com/>.
- [12] Controlled comfort (CoCo), dostopno na:
<http://www.coco-technology.com>.
- [13] XBMC media center, dostopno na:
<http://www.xbmc.org>.
- [14] Farnell, dostopno na:
<http://farnell.com/>.
- [15] Hobbyking, dostopno na:
<http://www.hobbyking.com>.
- [16] ILI9341 LCD library, dostopno na:
<http://stm32f4-discovery.com/2014/04/library-08-ili9341-lcd-on-stm32f429-discovery-board/>.
- [17] Elecrow, dostopno na:
<http://www.elecrow.com>.
- [18] Intertechno, dostopno na:
<http://intertechno.at/>.
- [19] 433MHz for Arduino, dostopno na:
<https://bitbucket.org/fuzzillogic/433mhzforarduino/>.